Studies in Computational Intelligence 1200

Oscar Humberto Montiel Ross
Ulises Orozco-Rosas
Anabel Martínez-Vargas *Editors*

Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1

Springer

Studies in Computational Intelligence

Volume 1200

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

Editorial Board

Marco Dorigo, Université Libre de Bruxelles, Bruxelles, Belgium

Andries Engelbrecht, University of Stellenbosch, Stellenbosch, South Africa

Vladik Kreinovich, University of Texas at El Paso, El Paso, TX, USA

Francesco Carlo Morabito, Mediterranea University of Reggio Calabria, Reggio Calabria, Italy

Roman Slowinski, Poznan University of Technology, Poznan, Poland

Yingxu Wang, Schulich School of Engineering, Calgary, AB, Canada

Yaochu Jin, Westlake University, Hangzhou, China

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI AG (Switzerland), zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

Oscar Humberto Montiel Ross ·
Ulises Orozco-Rosas · Anabel Martínez-Vargas
Editors

Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1

Springer

Editors
Oscar Humberto Montiel Ross
Research and Development Department
Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI)
Tijuana, Baja California, Mexico

Anabel Martínez-Vargas
Universidad Politécnica de Pachuca
Zempoala, Hidalgo, Mexico

Ulises Orozco-Rosas
CETYS Universidad
Tijuana, Baja California, Mexico

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-031-85613-6 ISBN 978-3-031-85614-3 (eBook)
https://doi.org/10.1007/978-3-031-85614-3

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

In this era of rapid technological advancement, the integration of artificial intelligence (AI) and quantum computing serves as a transformative force across various domains. *Artificial Intelligence and Quantum Computing: Early Innovations* emerges as a comprehensive guide and scholarly resource that aligns with the United Nations Sustainable Development Goals (SDGs). These goals, which aim to ensure peace and prosperity for both people and the planet, frame the innovative research showcased in this volume—research that promises to elevate the quality of life, foster sustainable practices, and enhance decision-making processes in complex environments.

Scientifically, the book explores cutting-edge research in AI and quantum computing, covering topics from hybrid quantum-classical computing models to advanced optimization techniques. These are pivotal for progressing towards more efficient and capable systems, ensuring that readers gain both theoretical and practical insights into the current landscape and future potential of these technologies.

The book holds significant relevance to society, providing key insights for researchers, practitioners, and policymakers in fields such as healthcare, agriculture, finance, and infrastructure. By introducing quantum-enhanced machine learning for medical diagnostics, it underscores the potential of AI and quantum computing to improve global health, directly supporting SDG 3's objectives.

Here are the chapters specifically aligned with SDG 3: Good Health and Well-being, reflecting the book's emphasis on enhancing healthcare through technological innovations:

- *Hybrid Quantum Convolutional Network for Spine Fracture Classification*: Details the use of hybrid quantum and classical convolutional networks to accurately classify spine fractures, enhancing diagnostic precision and patient care.
- *RSNA 2024: Multi-view Quantum Deep Learning Model for Degenerative Lumbar Spine Classification*: Discusses a cutting-edge quantum deep learning model that analyzes multiple views of lumbar spine images to diagnose degenerative conditions, improving outcomes and treatment strategies.

- *Enhancing Nocturnal Blood Pressure Detection through Interval Type-2 Fuzzy Classification Systems*: Explores the application of interval type-2 fuzzy logic systems to improve the accuracy of nocturnal blood pressure monitoring, crucial for managing cardiovascular health.
- *Glioma Segmentation in T1-Weighted MRI Images Using Deep Learning*: Highlights the use of deep learning techniques to segment glioma tumors in MRI scans, aiding in more precise and effective treatment planning.
- *Vision Transformers for the Segmentation of Patterns Associated to Pathology in Diabetic Retinopathy*: Examines how vision transformers are used to detect and segment pathological patterns in retinal images of diabetic retinopathy patients, which is vital for preventing vision loss.
- *Segmentation of High Quality Histopathological Micrographs Using a Genetic Algorithm and Otsu's Method*: A method that combines a genetic algorithm with Otsu's method efficiently segments high-quality histopathological micrographs, reducing computational demands. This technique improves the precision of medical image analysis, especially in detecting complex patterns in breast cancer histopathology.

These chapters demonstrate the application of advanced AI and quantum computing techniques in critical areas of medical diagnostics and treatment, showcasing their potential to significantly improve global health outcomes.

SDG 9 emphasizes the importance of building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. This book addresses these aspects through chapters that explore the following topics:

- *AI-based Algorithm for the Automatic Detection of Missing Solder Balls in Encapsulated Semiconductors*: Highlights advancements in semiconductor manufacturing, crucial for the technology sector, by using AI to improve quality control processes.
- *Membrane Pseudo-Bacterial Potential Field with GPU Acceleration for Mobile Robot Path Planning*: Discusses the development of novel algorithms for robotics, which play a significant role in automation and efficient path planning in various industrial applications.
- *Error Correction in Quantum Computing*: This chapter delves into the technical advancements necessary to make quantum computing a viable and reliable technology for the future of industry and computing.
- *Supervised Quantum Machine Learning*: Introduces applications of quantum machine learning that are shaping the next generation of industrial innovations and data analysis techniques.
- *Hybrid Quantum Genetic Algorithm with a Strongly Entanglement Recombination Process for Maximizing the Coverage Area of a Set of Unmanned Aerial Vehicles in Natural Disaster Locations*: Covers the use of hybrid quantum algorithms to optimize UAV deployment, crucial for disaster response and infrastructure inspection.

- *Review on Quantum Optimization of Type-1, Type-2, and Type-3 Fuzzy Systems*: Examines how quantum computing can enhance fuzzy systems, widely used in control systems and industrial applications.
- *Review on Quantum Optimization of Neural Networks and Fuzzy Systems*: This chapter further investigates the integration of quantum computing with neural networks and fuzzy systems to drive innovation in data processing and decision-making systems.
- *Finance in the Quantum Computing Era: A Survey from 2019 to 2024*: Provides insights into how quantum computing is set to revolutionize the financial industry through enhanced security and computational capabilities.
- *From Classical Challenges to Quantum Solutions: Grover's Algorithm for the N-Queens Problem*: Discusses the application of quantum algorithms to solve classical problems more efficiently, showcasing the potential for breakthroughs in computational complexity.
- *Cross-Platform Analysis of Quantum Circuit Simulators with State Collapse Mechanisms and Back-Action Tracking in C and Python*: This chapter evaluates tools essential for the development of quantum computing applications, contributing to the infrastructure necessary for its broader adoption.
- *Analysis of BB84 Quantum Key Distribution Protocol*: Addresses innovations in quantum cryptography, ensuring the security of communications in an increasingly digital world.
- *Benchmarking ML, DL, and Transformer Models for Document Classification*: Looks into the role of advanced machine learning models in optimizing document processing systems, promoting efficient and responsible data management.
- *Artificial Intelligence in Astrophysics: Analytical Enhancements*: Details AI applications in astrophysics, contributing to our understanding of climate systems and environmental phenomena through enhanced data analysis.
- *Towards a Fully Quantum Learning System: Quantum Architecture Search with Quantum Evolutionary Algorithms*: Explores the development of quantum computing architectures using evolutionary algorithms to enhance the performance and capabilities of quantum systems.
- *Roach Infestation Optimization Algorithm with Enhanced Performance Using Automatic Parameter Adaptation Based on Fuzzy Systems*: Details an innovative optimization algorithm inspired by roach social behaviors, improved through automatic fuzzy logic-based parameter adjustments.
- *From Containers to the Cloud, Scaling Event-driven Evolutionary Algorithms:* Discusses methodologies for scaling evolutionary algorithms from localized containers to cloud-based systems, enhancing their efficiency and adaptability.
- *Comparative Analysis of Machine Learning Algorithms for Fault Prediction Using Software Complexity and Object-Oriented Metrics*: Examines different machine learning techniques for predicting software faults, utilizing complexity and object-oriented metrics to evaluate their effectiveness.
- *Application of IMU Sensors and AI for Infrastructure Monitoring*: Describes the integration of Inertial Measurement Units (IMU) and artificial intelligence to enhance the monitoring and maintenance

SDG 11 highlights the need for sustainable urban planning and robust community services to enhance the quality of life. This theme is explored through chapters like *Analysis of Geographic Coordinates Initialization Methods for Unmanned Aerial Vehicles Localization*, which examines methodologies for enhancing UAV localization vital for urban planning and community services.

SDG 15 focuses on managing life on land sustainably, conserving and restoring terrestrial ecosystems, and halting biodiversity loss. The book addresses these issues through practical applications of new technologies, as discussed in chapters like *An Intelligent System Design for Automated Quality Control of Agricultural Produce Based on Computer Vision*, which discusses the use of computer vision in automating the quality control of agricultural produce, enhancing food security and sustainable agricultural practices.

The organization of this book is meticulously structured around the Sustainable Development Goals, serving as a guiding framework to demonstrate how advanced technologies can be leveraged to meet these global objectives. Each section of the book is dedicated to a specific SDG, highlighting relevant chapters that demonstrate technological innovations and their applications in addressing the challenges associated with each goal.

We conclude this preface by expressing our sincere gratitude to all the individuals who have assisted us during the review and compilation of this book. We also extend our thanks to our respective institutions and CONAHCYT for supporting our research activities.

Tijuana, Baja California, Mexico — Prof. Dr. Oscar Humberto Montiel Ross
Tijuana, Baja California, Mexico — Prof. Dr. Ulises Orozco-Rosas
Zempoala, Hidalgo, Mexico — Prof. Dr. Anabel Martínez-Vargas

Contents

SDG 3 Good Health and Well-Being

Hybrid Quantum Convolutional Network for Spine Fracture Classification 3
Lopez Daniel Alejandro, Montiel Oscar, Castillo Oscar, and Lopez-Montiel Miguel

RSNA 2024: Multi-view Quantum Deep Learning Model for Degenerative Lumbar Spine Classification 23
Lopez Daniel Alejandro, Montiel Oscar, Castillo Oscar, and Lopez Miguel

Enhancing Nocturnal Blood Pressure Detection Through Interval Type-2 Fuzzy Classification Systems 45
Ivette Miramontes and Patricia Melin

Glioma Segmentation in T1-Weighted MRI Images Using Deep Learning 57
Fernando Morquecho Lara, Oscar Montiel, Miguel López Montiel, and Moisés Sánchez Adame

Vision Transformers for the Segmentation of Patterns Associated to Pathology in Diabetic Retinopathy: A Literature Review and Clinical Relevance in Eye Fundus Images 77
B. L. López-Covarrubias, L. J. González-Zazueta, J. I. Nieto-Hipolito, and G. J. Avilés-Rodríguez

Segmentation of High Quality Histopathological Micrographs Using a Genetic Algorithm and Otsu's Method 97
F. J. Rojas-Pérez, E. G. Pérez-Pérez, Anabel Martínez-Vargas, and J. R. Conde-Sánchez

SDG 9 Industry, Innovation, Technology and Infrastructure

AI-Based Algorithm for the Automatic Detection of Missing Solder Balls in Encapsulated Semiconductors 121
Nestor A. Zermeño Campos, Julio C. Rodriguez Quiñonez, and Carlos A. Aguilar Avelar

Membrane Pseudo-Bacterial Potential Field with GPU Acceleration for Mobile Robot Path Planning 145
Ulises Orozco-Rosas, Kenia Picos, and Oscar Montiel

Error Correction in Quantum Computing 167
Jesús Yaljá Montiel-Pérez, Jordi Fabián González-Contreras, Erik Zamora, and Juan Humberto Sossa-Azuela

Supervised Quantum Machine Learning 193
A. Monroy-Azpeitia, J. Y. Montiel-Pérez, A. Maldonado-Romo, A. A. Arvizu-Velazquez, J. L. López-Bonilla, and S. Vidal-Beltrán

Hybrid Quantum Genetic Algorithm with a Strongly Entanglement Recombination Process for Maximizing the Coverage Area of a Set of Unmanned Aerial Vehicles in Natural Disaster Locations 209
Enrique Ballinas, Oscar Montiel, and Anabel Martínez-Vargas

Review on Quantum Optimization of Type-1, Type-2 and Type-3 Fuzzy Systems 235
Oscar Castillo

Review on Quantum Optimization of Neural Networks and Fuzzy Systems 245
Patricia Melin

Finance in the Quantum Computing Era: A Survey from 2019 to 2024 255
Luis Enrique Andrade-Gorjoux, César Castrejón-Peralta, Jorge Saavedra-Benavides, Jesús Yaljá Montiel-Pérez, and Eduardo Bustos-Farias

From Classical Challenges to Quantum Solutions: Grover's Algorithm for the N-Queens Problem 275
Oscar Montiel, Ulises Orozco-Rosas, Daniel López, and Moisés Sánchez

Cross-Platform Analysis of Quantum Circuit Simulators with State Collapse Mechanisms and Back-Action Tracking in C and Python 303
Paul O. Onoja, Seyi Tope Ogunji, Moises Sanchez-Adame, Oscar Montiel, Ushnik Nath, Lubna M Siddiqui, and Fernando Alonso Morquecho Lara

Performance Dynamics of BB84 QKD Protocol in Noisy Quantum Channels: A Simulation Study 325
Seyi Tope Ogunji, Moisés Sánchez-Adame, Oscar Montiel, and Marisela Martinez-Quiroz

Benchmarking ML, DL, and Transformer Models for Document Classification 347
Ricardo Rioda Santiago, Antonio Contreras Ortiz, Claudia Tona, and Miguel Lopez-Montiel

Artificial Intelligence in Astrophysics: Analytical Enhancements 375
Luis H. Sanchez, Carlos A. Guerrero, Juan J. Tapia, and Ana V. Ojeda

Towards a Fully Quantum Learning System: Quantum Architecture Search with Quantum Evolutionary Algorithms 397
Yoshio Rubio, Cynthia Olvera, and Oscar Montiel

Roach Infestation Optimization Algorithm with Enhanced Performance Using Automatic Parameter Adaptation Based on Fuzzy Systems 413
Enrique Lizárraga, Fevrier Valdez, Oscar Castillo, and Patricia Melin

From Containers to the Cloud, Scaling Event-Driven Evolutionary Algorithms 425
Mario Garcia-Valdez, Alejandra Mancilla, and Juan J. Merelo Guervós

Comparative Evaluation of Machine Learning Algorithms for Predicting Defects Using Object-Oriented and Software Complexity Metrics 437
Yaretxy Arely Pelayo Lomeli, Juan Pablo García Vázquez, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Jesus Eduardo Soto Vega, and Araceli Celina Justo López

Application of IMU Sensors and AI for Infrastructure Monitoring 459
M. Fernanda López-Barajas, Julio C. Rodríguez-Quiñonez, Wendy Flores-Fuentes, Jonathan J. Sanchez-Castro, Jorge Alejandro Valdez-Rodríguez, Moises J. Castro-Toscano, and Oleg Sergiyenko

SDG 11 Sustainable Cities and Communities

Analysis of Geographic Coordinates Initialization Methods for Unmanned Aerial Vehicles Localization 491
Gabriela Rodríguez-Cortés, Anabel Martínez-Vargas, MA Cosío-León, and Oscar H. Montiel-Ross

SDG 15 Life on Land

An Intelligent System Design for Automated Quality Control of Agricultural Produce Based on Computer Vision 507
Kenia Picos, Ulises Orozco-Rosas, and Kevin Ruiz

SDG 3 Good Health and Well-Being

Hybrid Quantum Convolutional Network for Spine Fracture Classification

Lopez Daniel Alejandro, Montiel Oscar, Castillo Oscar, and Lopez-Montiel Miguel

Abstract Osteoporotic spine fractures significantly impact adults over the age of 50, underscoring the need for advanced diagnostic tools to enhance their quality of life. While Convolutional Neural Networks (CNNs) have established their effectiveness in diagnosing such conditions, this study explores a novel approach by integrating quantum computing elements. We present a hybrid quantum convolutional neural network based on the ResNet18 architecture, tailored for multiclass classification of spine lesions from the VinDr-SpineXR dataset. This model incorporates a quantum circuit that mimics a classical neural network layer, enhancing feature extraction capabilities. In the training phase, the hybrid quantum model records accuracy, recall, precision, and F1 scores of 0.8274, 0.8309, 0.8262, and 0.8262, respectively, which represent decreases of 9.65%, 9.35%, 9.75%, and 9.75% compared to the classical model's performance of 0.9158, 0.9166, 0.9155, and 0.9155. However, on the test set, the hybrid model demonstrates a significant improvement, with increases in accuracy and recall by 3.13% each, and in F1 score by 2.20%. These results underscore the model's enhanced ability to generalize from small datasets and its potential to refine feature extraction processes for medical imaging, showcasing its advantages for practical applications where generalization is critical.

Keywords Spine fracture · Computer aided diagnosis · Deep learning · CNNs · Quantum machine learning · QNNs

L. Daniel Alejandro · M. Oscar (✉)
Instituto Politécnico Nacional—CITEDI, Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

L. Daniel Alejandro
e-mail: dlopez@citedi.mx

C. Oscar
Instituto Tecnológico de Tijuana, Tijuana, Baja California, Mexico
e-mail: ocastillo@tectijuana.mx

L.-M. Miguel
ITJ Labs, Tijuana, Baja California, Mexico
e-mail: milopez@citedi.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_1

1 Introduction

Osteoporosis is a chronic disease that significantly impacts the quality of life for millions worldwide, especially adults over the age of 50. Among the primary complications are Fragility Fractures (FF), which not only have severe economic implications but also contribute to morbidity and mortality rates [1]. According to the International Osteoporosis Foundation (IOF), approximately one in three women and one in five men over 50 will suffer an FF, commonly affecting the wrist, pelvis, hip, and spine [1]. Given that osteoporosis is a key cause of spinal lesions, it is essential to develop reliable methods for assessing spinal damage effectively.

The integration of quantum computing with machine learning, known as quantum machine learning (QML), combines the strengths of quantum mechanics with traditional learning algorithms, facilitating improved performance in complex feature representation [2]. Hybrid algorithms that leverage quantum principles have shown promise in capturing intricate data patterns, offering advantages over classical models [3].

In response to these challenges, this study proposes a hybrid quantum convolutional neural network based on the classical ResNet18 architecture for multiclass classification of spine lesions in the VinDR-SpineXR dataset [4]. To address dataset imbalance, the dataset was undersampled to ensure equal class representation using unique images, presenting a small dataset challenge for both the classical and quantum models implemented. The hybrid quantum model combines convolutional and pooling operations within a quantum circuit that functions as a model layer with trainable parameters, allowing parameter optimization for enhanced training.

For the training set, the hybrid quantum model achieved accuracy, recall, precision, and F1 scores of 0.8274, 0.8309, 0.8262, and 0.8262, respectively, with decreases of 9.65%, 9.35%, 9.75% and 9.75% compared to the classical model's performance of 0.9158, 0.9166, 0.9155, and 0.9155. Notably, on the test set, the hybrid quantum model outperformed the classical model, showing increases in accuracy and recall by 3.13% each and an F1 score improvement of 2.20%. These findings indicate a trade-off in performance, with the hybrid model excelling in generalization and fine feature extraction in cases where small datasets are a limitation.

This chapter is organized as follows: Sect. 1 provides a brief introduction to spine fracture diagnosis and deep learning methods, while Sect. 2 discusses foundational concepts relevant to this work. Section 3 describes the implemented methodology, and Sect. 4 presents the experiments, results, and discussion. Finally, Sect. 5 offers conclusions and highlights future directions for research in this area.

2 Fundamentals

2.1 Convolutional Neural Networks

Since their emergence in 1998 [5], CNNs have proven to be reliable deep learning (DL) methods with the ability to learn complex features in an automated manner, enhancing feature abstraction when working with unstructured datasets and enabling solutions to difficult tasks such as speech, text, and image recognition. These applications have impacted healthcare systems, autonomous vehicles, cybersecurity, and predictive analytics [6]. CNNs are comprised of three main sections: the **input layers**, where input data enters to be processed; **hidden layers**, where key feature extraction occurs; and **output layers**, which produce a probability result based on the generalizations obtained from the input data features. The defining characteristic of CNNs is the **convolutional operation**, which relies on trainable parameters and focuses on extracting relevant features for specific tasks [7].

The convolutional layer utilizes a small subset of values defined as *kernels*, which are applied to the input data in a *tensor* format. This layer processes each part of the input by multiplying and adding each element to form new feature maps [8]. Expressed mathematically, if the convolutional layer applies a filter w_k to a region of the input, denoted as X at every position (i, j), the convolutional layer output can be written as:

$$Z_{i,j,k} = \sum_{m} \sum_{n} w_{m,n,k} \cdot X_{i+m,j+n} + \theta_k, \tag{1}$$

where the output $w_{m,n,k}$ represents the weight at position (m, n) and θ_k is the bias for the k-th filter. To enhance feature extraction for nonlinear features, CNNs incorporate **activation functions**, which normalize input data to a range to a range suitable for computation by neurons [9]. The computation of an activation map on a convolutional feature map can be expressed as:

$$a_{i,j,k} = a(Z_{i,j,k}). \tag{2}$$

Another essential layer is the **pooling layer**, which reduces the spatial dimensions of the input data by computing the average or maximum of an established size, denoted as $p \times q$ [10]. For instance, computation in a max pooling layer is defined as:

$$P_{i,j,k} = \max_{m=0}^{p-1} \max_{n=0}^{q-1} Z_{i+m,j+n,k}, \tag{3}$$

where $Z_{i,j,k}$ represents the input value of the pooling layer and $p \times q$ is the size of the pooling window. Following feature extraction in hidden layers, CNNs utilize **fully connected** layers, which combine all features from the previous layers to a single

vector and apply linear transformation to prepare model output [10]. For an output y, a weighted sum of input Z and its associated bias θ is calculated as follows:

$$y = \sum_k w_k \cdot Z_k + \theta \tag{4}$$

In image classification, once the model provides an output, performance is assessed by calculating the **loss** and an **objective function**. The model's loss refers to the discrepancy in predictions, indicating the proportion of predicted labels that match the target labels. Additionally, the average loss over the entire dataset is obtained through the **cost function** [11]. For a typical loss computation such as the Cross-Entropy loss function, this is written as:

$$L(y, \hat{y}) = -\sum_c y_c \log(\hat{y}_c), \tag{5}$$

where y is the true label for the target, $\hat{y}$ is the predicted output, y_c is the true label for class c, and $\hat{y}_c$ is the predicted probability for class c. After calculating the loss, the cost function $J(W, \theta)$, which represents the mean loss for N samples in the dataset, is given by:

$$J(W, \theta) = \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, \hat{y}^{(i)}; W, \theta). \tag{6}$$

2.2 *Quantum Computing*

Quantum computing uses the basic information unit known as the *qubit*, which can represent two basis states simultaneously through the principle of **superposition** in quantum mechanics. The most common basis states are $|0\rangle$ and $|1\rangle$ which are analogous to the classical "0" and "1" bits [12]. In a two-term combination of $|0\rangle$ and $|1\rangle$, the complex numbers α and β to represent the probability amplitudes for each state, expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{7}$$

As with all probabilities, this expression must satisfy the normalization condition of $|\alpha|^2 + |\beta|^2 = 1$, where all possible outcomes add up to 1. The **Bloch sphere** is commonly used to represent superposition states, providing a three-dimensional visualization of all potential quantum states and illustrating how a qubit state fits within this space, see Fig. 1.

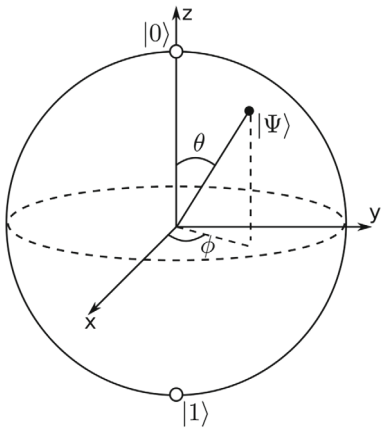


Fig. 1 Bloch sphere diagram

In the Bloch sphere, the states $|0\rangle$ and $|1\rangle$ align with the the z-axis, corresponding to the classical bit values of "0" and "1". The vector Ψ represents a qubit in superposition, with angles θ and ϕ determining its orientation on the sphere. Considering θ as the angle between ψ and the z-axis, and ϕ as the phase of superposition in the x-y plane, this state can be mathematically represented as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle . \tag{8}$$

Similar to classical computing, combining multiple qubits forms a composed quantum system, which describes the interactions among qubits within the system or with external systems [13]. A quantum system Ψ comprised of N qubits is written as

$$|\Psi\rangle = |\psi_1\rangle \otimes ... \otimes |\psi_n\rangle . \tag{9}$$

where the composite system is denoted by the tensor product of each qubit. To facilitate quantum circuit design, quantum systems are often written in base-2 notation as $|\Psi\rangle = c_0\,|00\ldots0\rangle + c_1\,|00\ldots1\rangle + \cdots + c_{2^n-1} = \sum_{i=0}^{2^n-1} c_i\,|i\rangle_2$.

To manipulate qubits, quantum gates are used, which transform their states. Quantum gates operate similarly to classical gates in digital systems but adhere to quantum mechanics, enabling state manipulation [14]. A quantum gate U acting on a quantum system operates as follows:

$$U\,|\Psi\rangle = U\sum_{i=0}^{2^n-1}\alpha_i\,|i\rangle , \tag{10}$$

where U represents the quantum gate, and $\sum_{i=0}^{2^n-1}$ refers to the probability amplitudes α for the possible states $|i\rangle$ [13].

A common quantum gate is the **Pauli-X gate**, represented by the matrix

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{11}$$

This gate performs a bit-flip operation, switching the qubit's state from $|0\rangle$ to $|1\rangle$ and from $|1\rangle$ to $|0\rangle$. It is analogous to the classical NOT gate and acts on a single qubit. In contrast, the CNOT gate (Controlled-NOT gate) is a two-qubit gate. It flips the state of the target qubit only when the control qubit is in the state $|1\rangle$. The matrix representation of the CNOT gate is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{12}$$

Another example, the Hadamard gate, creates superposition in the qubit it is applied to and is represented as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{13}$$

After quantum processing, **measuring** the quantum system allows for retrieving information in the classical state [13]. Quantum measurement can be described using the spectral decomposition of an observable M, where the observable M is a Hermitian operator. This operator can be decomposed into a sum of its eigenvalues m and corresponding projection operators P_m. The eigenvalues represent the possible outcomes of the measurement, and the projection operators P_m correspond to the measurement outcomes in the quantum state space. The spectral decomposition is expressed as:

$$M = \sum_m m P_m, \tag{14}$$

where P_m are the projection operators such that $P_m P_{m'} = \delta_{mm'} P_m$, and they satisfy the completeness relation $\sum_m P_m = I$ (the identity operator).

3 Methodology

In this section, the methodology employed to achieve spine lesion classification through classical deep learning and quantum deep learning techniques is detailed. The presented approach integrates a classical ResNet18 architecture with a quantum circuit to improve the accuracy of medical image diagnosis. The included subsections elaborate on the dataset used, deep learning model, quantum circuit, and its integration into the hybrid quantum model applied in this work.

3.1 Dataset

The dataset employed is *VinDr-SpineXR* [4], a spine x-ray dataset composed of 12,000 spine lesion samples categorized into 8 classes: Disc space narrowing, foraminal stenosis, no finding, osteophytes, other lesions, spondylolisthesis, surgical implant, and vertebral collapse. Each spine lesion x-ray image is classified into three types of radiographic images according to the study and series identifier. Additionally, each x-ray image includes bounding boxes marking the regions of interest for various types of spine lesions, with some images displaying multiple lesions from the same or different classes [4]. Due to this, only unique image entries are considered, reducing the dataset to 8,300 x-ray images. Figure 2, illustrates the data distribution for the three types of radiographic images in the dataset. It is evident that the majority classes across the three distributions are "No finding" and "Osteophytes", with more than 1,000 samples each. In contrast, the remaining minority classes have fewer than 500 samples, necessitating a downsampling process to standardize the sample size to 233 per class. However, classes such as disc space narrowing, vertebral collapse, and surgical implant do not reach this number.

To address data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) [15] is utilized to balance the minority classes and achieve a total of 233 samples per class. This augmentation method generates new samples for the minority classes by interpolating data points from the existing samples. For image

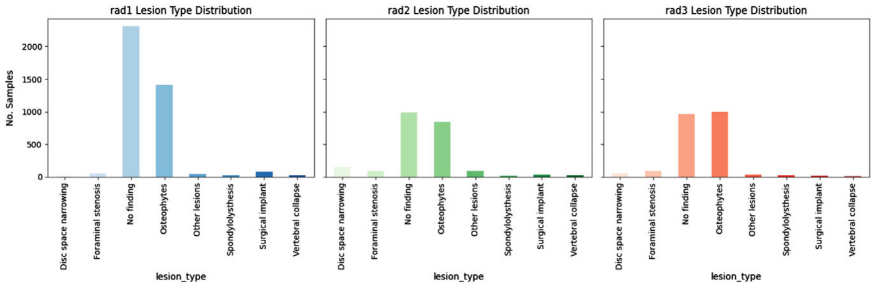


Fig. 2 Data distribution for the VinDr-SpineXR dataset

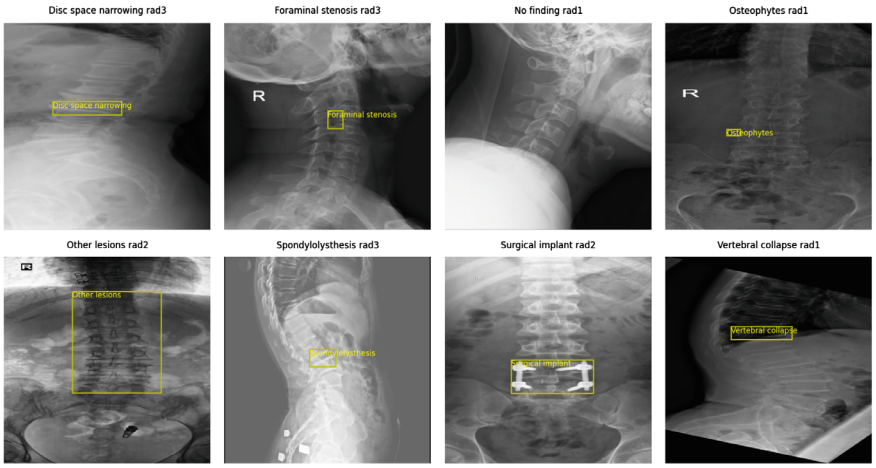


Fig. 3 Spine lesion classes and corresponding bounding boxes

data, this technique can be tailored by applying it to small image patches, taking into account pixel intensity and color channels. Figure 3 depicts all spine lesion classes, with bounding boxes highlighting the region of interest. It is important to note that these data points were not specifically used in this study, as the primary aim was to evaluate classification performance without additional data preparation techniques such as image cropping.

3.2 Classic Deep Learning Model

The selected DL model is ResNet18 [16], a residual convolutional neural network (CNN) composed of convolutional layers, organized in distinct **residual blocks**. The key feature of this model is its approach to residual learning, which mitigates vanishing gradients and reduces the risk of overfitting. The ResNet architecture introduces residual connections that skip one or more layers. This structure allows the network to bypass certain layers and facilitates gradient flow during training [16]. Compared to its larger counterparts, ResNet18 consists of only 18 layers, making it a medium-sized but effective CNN capable of handling complex image classification problems. The operation of a residual block can be expressed mathematically as:

$$y = F(x, W) + x, \tag{15}$$

where the output y is computed from the input x; $F(x, W)$ represents the mapping function of previous layers, and W dennotes the weights. This equation illustrates how the input x is added to the resulting feature maps, allowing it to bypass the processing

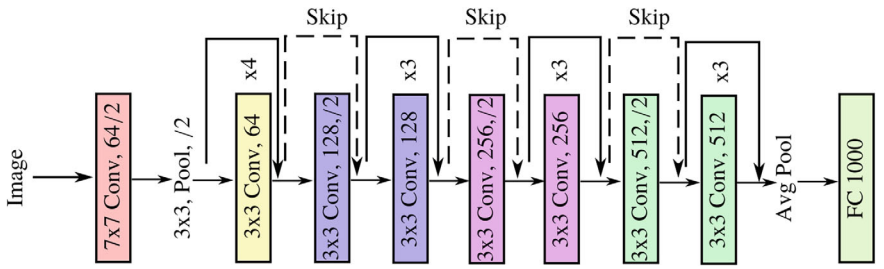


Fig. 4 ResNet18 deep learning model architecture. Adapted from [16]

of previous layers [16]. Figure 4 presents the ResNet18 architecture, highlighting the residual and skip connections as data dimensions are reduced and the number of filters increases.

The convolutional layers depicted in Fig. 4 that maintain input dimensions are repeated three times. With the exception of the initial convolution with 64 filters, all convolutional blocks begin with a skip connection from the preceding block.

3.3 Hybrid Quantum Multi-view Model

The proposed hybrid quantum model integrates the ResNet18 architecture with a quantum circuit that functions as a feature extraction layer at the end of the architecture. The quantum circuit is designed using the PennyLane library [17] and is composed of quantum gates that operate similarly to a convolutional layer and a pooling layer. To achieve this, classical data output from previous layers is encoded into a quantum state for processing; this is done through parameterized rotations on the y-axis using the AngleEmbedding operator, which consists of RY gates encoding data for further quantum operations. Once classical data is encoded, quantum feature extraction can commence, with the two sets of quantum gates divided as follows:

- **Quantum convolutional layer**: After encoding the classical data, a series of RY and CNOT gates mimics the function of a classical convolutional layer by extracting significant input features. Rotations on the y-axis influence the control qubits and their neighboring qubits, altering their states and creating entanglement. This enables feature extraction by establishing correlations between input features. A second quantum convolutional layer is applied after the quantum pooling layer for final feature extraction.
- **Quantum pooling layer**: A combination of RY, CNOT, and Hadamard gates is used after the quantum convolutional layer to perform quantum pooling. Hadamard gates introduce superposition, and together with CNOT gates, simulate the behavior of a classical average pooling layer by averaging out the input data from previous gates.

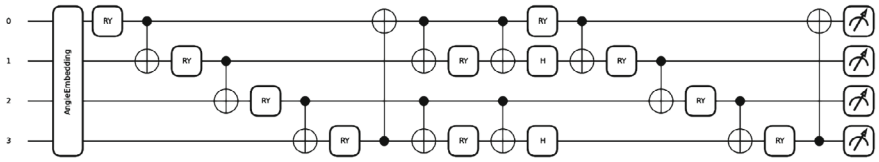


Fig. 5 Proposed quantum circuit

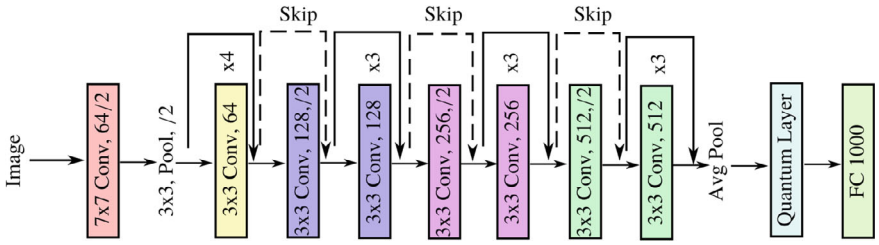


Fig. 6 Proposed hybrid quantum model based on ResNet18 architecture

After completing quantum processing, the quantum state is measured and converted back to classical data for further analysis. This process, known as **decoding**, is achieved by measuring the quantum state along the z-axis, collapsing the qubit states and enabling subsequent processing and model output. The proposed quantum circuit, shown in Fig. 5, consists of four qubits to match the data dimensions at the end of the ResNet18 architecture.

With the quantum circuit defined, the hybrid quantum architecture incorporates this circuit as a trainable layer within the classical ResNet18 model for final feature extraction with an input size of (4, 4). Figure 6 illustrates the hybrid quantum model, where eight trainable parameters are optimized in each training epoch, tuning the quantum gate parameters to enhance model performance.

3.4 Classification Metrics

The proposed models are evaluated using the classification metrics of accuracy (ACC), precision (P), recall (R), and F1 score (F1). To compute these metrics, the predicted values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are essential. TP and TN values represent correctly predicted positive and negative samples, indicating that they belong to the evaluated class. Meanwhile, FP and FN values denote samples incorrectly classified as part of the evaluated class [18]. Model predictions can be visually represented through a confusion matrix, which is a diagram composed of cells illustrating the number of predicted values for each of the classes of the dataset. Figure 7 displays a multiclass confusion matrix, used for problems with more than two classes. In this representa-

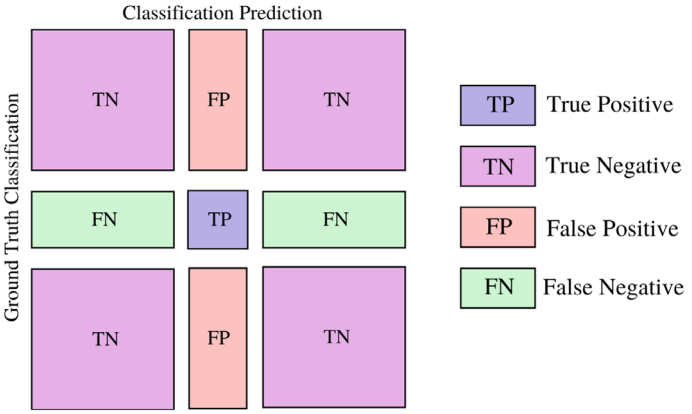


Fig. 7 Multiclass confusion matrix

tion, all predicted values are compared to the correct labels of the dataset, where the main diagonal comprises TP values, indicating correctly predicted samples for each class [18].

Once the model's predicted values are compiled, classification metrics such as accuracy, precision, recall, and F1 score can be calculated. TThe ACC metric represents model performance across all samples in the dataset, considering correctly predicted positive and negative instances. Precision (P) and recall (R) refer to the model's ability to correctly predict positive samples out of all positive samples. The F1 score summarizes model performance, and displays the balance between P and R to assess correct model predictions.

$$\begin{aligned} P &= \frac{TP}{TP + FP}, \\ R &= \frac{TP}{TP + FN}, \\ ACC &= \frac{TP + TN}{TP + TN + FP + FN}, \\ F1 &= 2\left(\frac{P \times R}{P + R}\right). \end{aligned} \tag{16}$$

In addition to the confusion matrix, the Area Under the Receiver Operating Characteristic Curve (AUROC) is employed to evaluate performance on the test set for both proposed models. The ROC curve illustrates classification performance for each class in the dataset across all possible thresholds [19]. The ROC curve is derived from the True Positive Rate (TPR) and False Positive Rate (FPR). These metrics represent the proportion of correctly and incorrectly predicted positive and negative samples, respectively, and are calculated as follows:

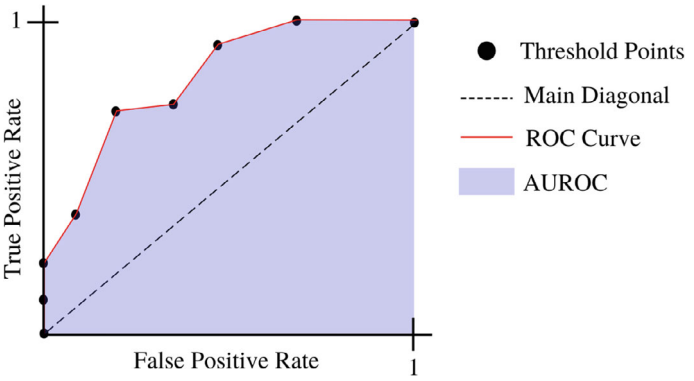


Fig. 8 AUROC curve example

$$FPR = \frac{FP}{FP + TN}$$
$$TPR = \frac{TP}{TP + FN}. \tag{17}$$

After obtaining the FPR and TPR values, the ROC curve can be generated. The example ROC curve in Fig. 8 includes two curves: the main diagonal and the ROC curve obtained from threshold points. The main diagonal serves as a baseline, representing a model performance of 50%, indicating that curves above it outperform a random classifier. Each threshold point on the curve corresponds to the minimum probability value required for a sample to be classified as positive; otherwise, it is classified as negative.

Once all possible thresholds have been evaluated, the ROC curve is completed. This process is analogous to calculating all potential confusion matrices for each threshold value and can be expressed mathematically as:

$$ROC(\sigma) = (FPR(\sigma), TPR(\sigma)), \tag{18}$$

where FPR and TPR are assessed for each given threshold σ [20]. Once the ROC curve is plotted, the Area Under the ROC (AUROC) can be computed, shown by the shaded region in Fig. 8. The AUROC for a given σ threshold is calculated as:

$$AUROC(\sigma) = \int_a^b TPR(\sigma) d(FPR(\sigma). \tag{19}$$

For multiclass problems, the AUROC curves used are "One versus Rest" (OvR), comparing each evaluated class against all other classes by treating one as the positive class and the remainder as negative. An OvR score function expressed as

$f = (f^1, \ldots, f^{N_c})$ forms an AUC score for each f^i, where the i-th class is positive, and all other classes j are negative [20]. The AUROC OvR metric is computed as:

$$AUROC^{OvR}(f) = \frac{1}{N_C} \sum_{i=1}^{N_C} AUROC_{i|\neg i}(f^i). \tag{20}$$

4 Experiments and Results

The experiments were conducted on an Ubuntu 22.04 system with CUDA toolkit 12.5 using the Pytorch framework with torch version 2.0.1. Quantum circuit simulations were performed using the PennyLane 0.37 library. The setup included an AMD Ryzen Threadripper 1920X CPU and an NVIDIA RTX 3080 GPU. The ResNet18 architecture, pretrained on the ImageNet1K_V2 dataset [21], was utilized. For model optimization, a balanced focal loss function and the Adam optimizer [22] were used. The Grid Search optimization method [23] was implemented to determine the optimal training configuration, tuning hyperparameters such as frozen layers, learning rate, and weight decay to achieve the best F1 score through 3-fold cross-validation. The optimized hyperparameters were a learning rate of 1×10^{-4}, no frozen layers, and a weight decay of 1×10^{-5}, a batch size of 32, and 10 training epochs with a patience of 5.

Experiments Overview: The experimental framework poses significant challenges due to the undersampled dataset, which presents data imbalances and high feature complexity, complicating multiclass classification. We conducted four experiments to evaluate the performance of both the classical and hybrid quantum models. The first and second experiments focused on training these models using classification metrics such as accuracy, precision, recall, and F1 score. The third and fourth experiments were dedicated to evaluating performance on the test set using additional metrics, including the multiclass confusion matrix and AUROC curves.

The first experiment aimed to train the classical ResNet18 architecture. As depicted in Fig. 9, the classical model exhibited a significant disparity between training and validation results, achieving an accuracy of 0.92 on the training set but only 0.52 on the validation set. This discrepancy, evident in the loss curves, suggests overfitting, leading to suboptimal performance. Furthermore, owing to the implementation of the Grid Search optimization method, changes in model training performance were minimal, as the selected hyperparameter combination quickly achieved the highest F1 score, outperforming other configurations from the early epochs.

Similar trends were noted for recall, precision, and F1 scores as illustrated in Fig. 10. The model achieved training and validation precision, recall, and F1 scores of approximately 0.90 and 0.55 across all metrics, respectively.

The second experiment was centered on the training process of the hybrid quantum model, and as seen in Fig. 11, the attained results are comparable but slightly inferior

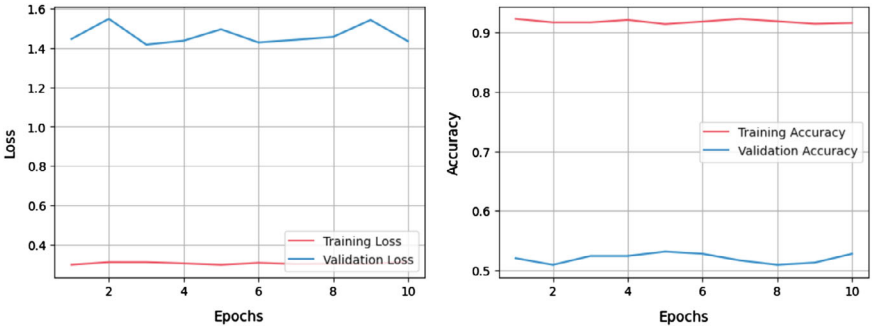


Fig. 9 Accuracy and loss curves for classical model

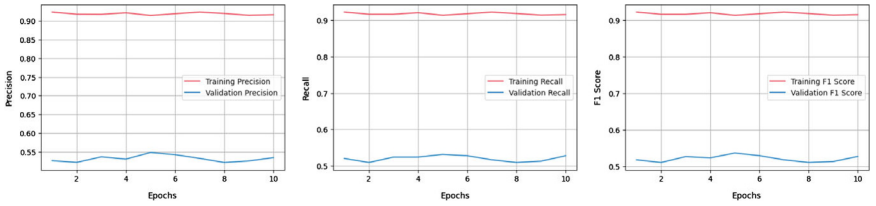


Fig. 10 Recall, precision, and F1 score curves for classical model

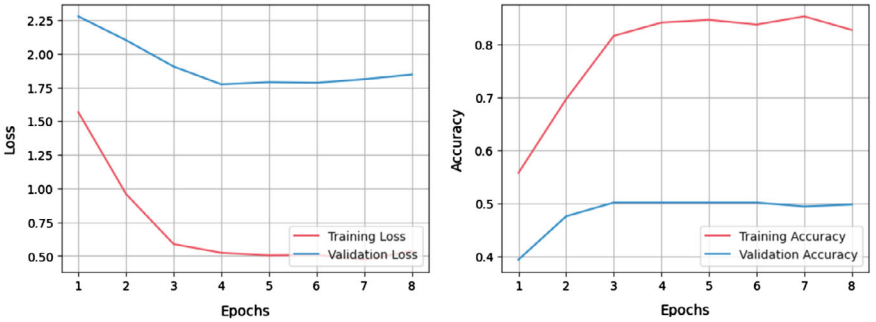


Fig. 11 Accuracy and loss curves for hybrid quantum model

to its classical counterpart, reaching an accuracy of 0.82 and 0.50 for train and validation sets respectively. However, a main difference to its classical counterpart is the training curves behavior, where the quantum layer weights adapt to the optimal value and stay there after epoch 4.

The rest of the classification metrics present similar results, attaining 0.83 and 0.50 for both the train and validation sets as seen in Fig. 12. The similar behavior for this set of curves to the ones in Fig. 11 demonstrates that although the hybrid quantum model underperforms when compared to its classical counterpart for the training set, it still attains comparable results for unlabeled data such as the validation set.

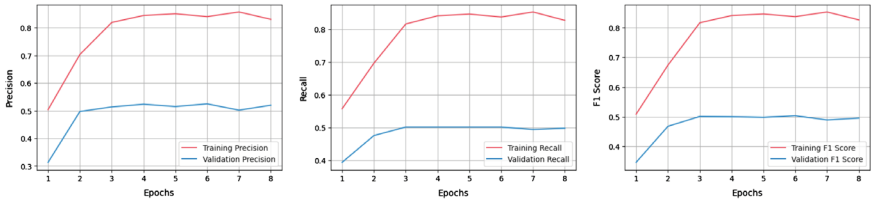


Fig. 12 Recall, precision, and F1 score curves for hybrid quantum model

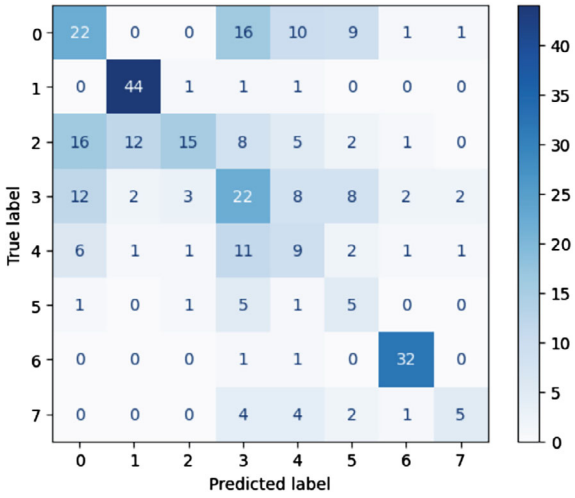


Fig. 13 Multiclass confusion matrix of the classical model for the test set

Furthermore, the quantum layer allows for additional tuning, despite not achieving higher results.

After model training is concluded, model inference is evaluated through the multiclass confusion matrix and AUROC curves.

The third experiment focuses on evaluating the test set results of the classical ResNet18 model, which presents a similar data imbalance to the training set. However, as the intention is to evaluate performance on real samples, neither data augmentation nor balancing techniques are considered. Observing the obtained confusion matrix of Fig. 13, the classical model attains the best results for class 1, 3, and 6, while still correctly predicting some samples of the minority classes. As test data distribution is imbalanced, classes with fewer samples such as 4, 5, and 7 are under represented and more difficult.

The AUROC curves further illustrate model performance for every class as shown in Fig. 14. The obtained AUROC values demonstrate that the model achieves its best performance for classes 1 and 6. However, the AUROC for minority classes 5 and 7 is relatively high, despite having fewer samples.

Evaluation of the hybrid quantum model inference is conducted in the fourth experiment. The attained results are comparable to its classical counterpart, show-

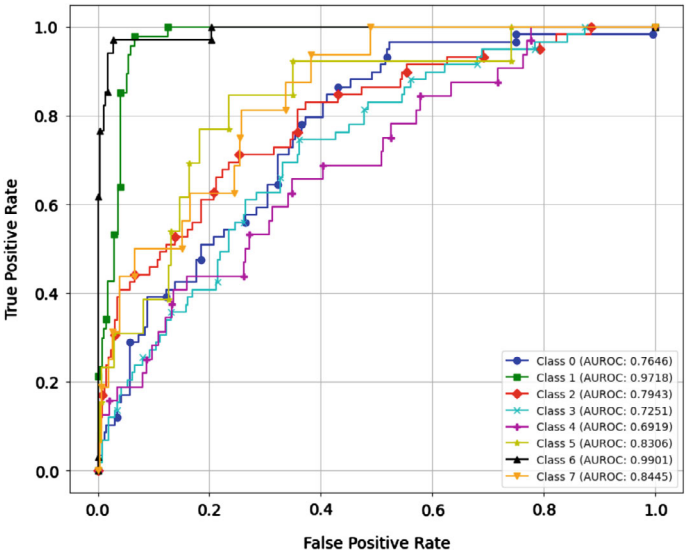
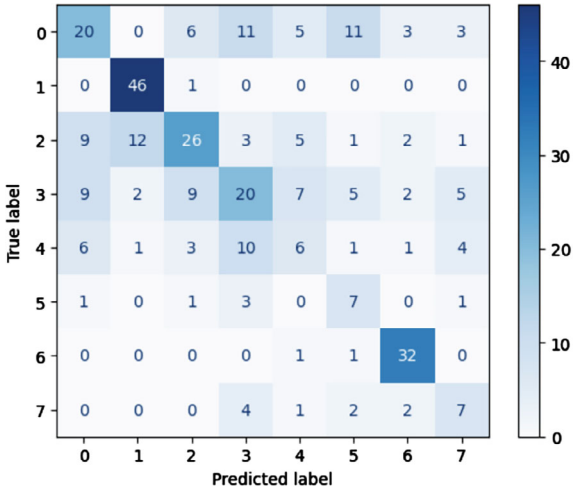


Fig. 14 Classical model AUROC curves for the test set

Fig. 15 Multiclass confusion matrix of the hybrid quantum model for the test set



casing a trade-off for classes 1 and 2, significantly increasing the number of correctly predicted samples for class 2. Furthermore, minority classes such as 5 and 7 show a slight improvement, as seen in Fig. 15. This demonstrates that the hybrid quantum model can effectively extract features for minority classes while slightly compromising performance for majority classes.

Although the matrix confusion results for the hybrid quantum model favor minority classes, the plotted AUROC curves in Fig. 16 shows slight underperformance

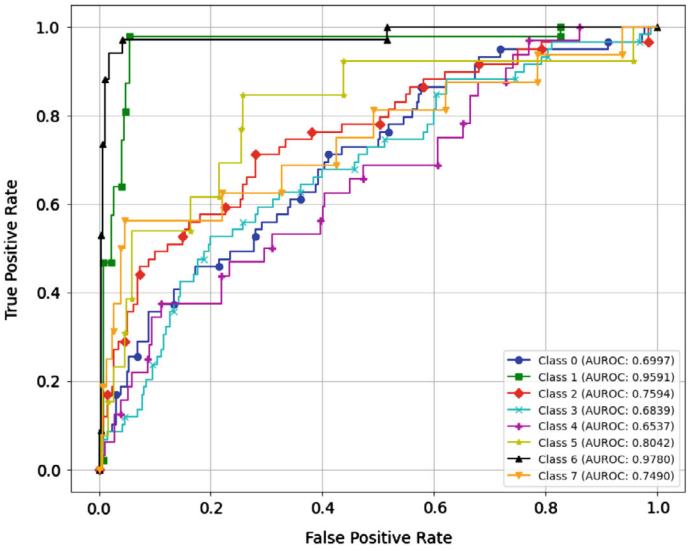


Fig. 16 Hybrid quantum model AUROC curves for the test set

Table 1 Proposed classification model comparison. **C** stands for classical, **HQ** stands for Hybrid Quantum. Best results in bold

Model	Train				Validation				Test				
	ACC	P	R	F1	ACC	P	R	F1	ACC	P	R	F1	AUC
C	**0.9158**	**0.9166**	**0.9158**	**0.9155**	**0.5279**	**0.5344**	**0.5279**	**0.5273**	0.4828	**0.5213**	0.4828	0.4761	**0.8266**
HQ	0.8274	0.8309	0.8274	0.8262	0.4981	0.5200	0.4981	0.4954	**0.5141**	0.5026	**0.5141**	**0.4993**	0.7859

when considering different threshold values. The attained AUROC values are comparable but slightly inferior to the ones obtained by its classical counterpart, struggling with classes 0, 3, and 4.

The classification results for both models are compiled in Table 1, where the classical model outperforms on all classification metrics for the train set. However, the results are more comparable for unlabeled data in the validation and test sets, with the hybrid quantum model occasionally outperforming in metrics such as accuracy, recall, and F1 score on the test set. These results suggest that although generalization for the training set is lower, the hybrid quantum model demonstrates better results on the most objective set of samples, as it is able to outperform when identifying new, never-seen spine lesions; showcasing its fine feature extraction capabilities when compared to its classical counterpart.

5 Conclusions and Future Work

As one of the leading causes for spine lesions are fragility fractures originated from osteoporosis, it is crucial to aid in the development of methods that help in their early diagnosis, prevention, and identification. Because of this, this work focuses on developing a quantum convolutional neural network based on the ResNet18 architecture for spine lesion multiclass classification of the VinDr-SpineXR dataset.

The proposed methodology is challenged by undersampling the dataset samples to balance class relevancy, and using the SMOTE augmentation technique to increase minority class samples, ending with a total of 233 spine lesion images per class considering artificial samples. The hybrid quantum neural network is built by adding a quantum layer at the end of feature extraction, comprised of a quantum circuit that works analogously to a classical block that encompasses two convolutional layers, and a pooling layer. The proposed quantum convolutional layer uses a series of RY, and CNOT gates to produce correlation between the circuit's qubits to extract relevant features, while the quantum pooling layer is generated by including a Hadamard gate to average out the generated feature maps by using a superposition transformation. To attain the best training hyperparameters, a 3-fold Grid Search optimization is implemented to tune the number of frozen layers, weight decay, and learning rate to use for both models.

The experimentation phase is split into the training and evaluation experiments for both models, where the proposed hybrid quantum model achieves an accuracy, precision, recall, and F1 score of 0.8274, 0.8309, 0.8274, 0.8262, respectively, which represent decreases of 9.65%, 9.35%, 9.75%, and 9.75% compared to the classical model's performance of 0.9158, 0.9166, 0.915, and 0.9155. Contrarily, during the testing phase, the hybrid model exhibited notable improvements, with increases in accuracy and recall by 3.13% each, and in F1 score by 2.20%. These results underscore the model's enhanced ability to generalize from small datasets and refine feature extraction processes, illustrating its practical advantages in settings where such generalization is crucial.

In conclusion, the hybrid quantum model has shown potential in harnessing quantum approaches and circuits for intricate feature extraction in limited datasets for spine lesion classification. The ability to improve performance on minority classes while marginally affecting majority class outcomes suggests its suitability for real-world applications.

Future work may include expanding the dataset for minority classes and utilizing provided bounding boxes to focus on smaller lesion features. Further exploration into deeper and more sophisticated classification architectures and quantum circuits could also yield additional improvements, enhancing the model's diagnostic capabilities.

Acknowledgements This research has received funding from the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) through project CF-2023-I-108, and from Instituto Politécnico Nacional (IPN), under the Research Project SIP20240164.

References

1. Clark, P., Caló, M., Torres-Naranjo, J.F., Cisneros-Dreinhofer, F., Silveira-Torre, L.H., Tapia-Hernández, M., Medina-Chávez, J.H., Gutiérrez-Robledo, L.M., Reza-Albarrán, A.A., Coronado-Zarco, R., León, A.O.G., Soria-Bastida, M., Islas-Upegui, M.M., Tejeda-Chávez, E.S., López-Cervantes, R.E., Jiménez-Herrera, B.L.: Osteoporosis and fragility fractures in mexico: A call to action. Archives of Medical Research 55, 103062 (2024). https://doi.org/10.1016/J.ARCMED.2024.103062
2. Wei, L., Liu, H., Xu, J., Shi, L., Shan, Z., Zhao, B., Gao, Y.: Quantum machine learning in medical image analysis. Neurocomputing 525, 42–53 (2023). https://doi.org/10.1016/J.NEUCOM.2023.01.049
3. Matondo-Mvula, N., Elleithy, K.: Advances in quantum medical image analysis using machine learning: Current status and future directions. Proceedings - 2023 IEEE International Conference on Quantum Computing and Engineering, QCE 2023 1, 367–377 (2023). https://doi.org/10.1109/QCE57702.2023.00049
4. Nguyen, H.T., Pham, H.H., Nguyen, N.T., Nguyen, H.Q., Huynh, T.Q., Dao, M., Vu, V.: Vindr-spinexr: A deep learning framework for spinal lesions detection and classification from radiographs. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12905 LNCS, 291–301 (2021). https://doi.org/10.48550/arxiv.2106.12930
5. Yann, L., Léon, B., Yoshua, B., Patrick, H.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2323 (1998). https://doi.org/10.1109/5.726791
6. Shiri, F.M., Perumal, T., Mustapha, N., Mohamed, R.: A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru (2023).
7. Paheding, S., Saleem, A., Siddiqui, M.F.H., Rawashdeh, N., Essa, A., Reyes, A.A.: Advancing horizons in remote sensing: a comprehensive survey of deep learning models and applications in image classification and beyond. Neural Computing and Applications 2024 36:27 36, 16727–16767 (2024). https://doi.org/10.1007/S00521-024-10165-7
8. Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights into Imaging 9, 611–629 (2018). https://doi.org/10.1007/S13244-018-0639-9/FIGURES/15
9. Dubey, S.R., Singh, S.K., Chaudhuri, B.B.: Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing 503, 92–108 (2021). https://doi.org/10.1016/j.neucom.2022.06.111
10. Krichen, M.: Convolutional neural networks: A survey. Computers 2023, Vol. 12, Page 151 12, 151 (2023). https://doi.org/10.3390/COMPUTERS12080151
11. Janocha, K., Czarnecki, W.M.: On loss functions for deep neural networks in classification. Schedae Informaticae 25, 49–59 (2017). https://doi.org/10.4467/20838476SI.16.004.6185
12. Cui, R., Lyu, Z.: Analysis of quantum gates in quantum circuits. Theoretical and Natural Science 10, 1–8 (2023). https://doi.org/10.54254/2753-8818/10/20230301
13. Zheng, J., Gao, Q., Lü, J., Ogorzałek, M., Pan, Y., Lü, Y.: Design of a quantum convolutional neural network on quantum circuits. Journal of the Franklin Institute (2022). https://doi.org/10.1016/J.JFRANKLIN.2022.07.033
14. Ross, O.H.M.: A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. IEEE Access 8, 814–838 (2020). https://doi.org/10.1109/ACCESS.2019.2962155
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002).
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December, 770–778 (2015). https://doi.org/10.1109/CVPR.2016.90
17. Pennylane: Automatic differentiation of hybrid quantum-classical computations (2018). https://doi.org/10.48550/arxiv.1811.04968

18. Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview (2020). https://doi.org/10.48550/arxiv.2008.05756
19. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves, vol. 06, pp. 233–240 (2006). https://doi.org/10.1145/1143844.1143874
20. Villardón, G., Zhao, X., Le, P.B., Nguyen, Z.T.: Roc curves, loss functions, and distorted probabilities in binary classification (2022). https://doi.org/10.3390/math10091410
21. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
22. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2014). https://doi.org/10.48550/arxiv.1412.6980
23. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research 13(Feb), 281–305 (2012)

RSNA 2024: Multi-view Quantum Deep Learning Model for Degenerative Lumbar Spine Classification

Lopez Daniel Alejandro, Montiel Oscar, Castillo Oscar, and Lopez Miguel

Abstract Spine lesions are one of the leading causes of disability worldwide, often caused by degenerative changes in the intervertebral disc spaces that harm spinal nerves. Deep Learning tools such as Convolutional Neural Networks have demonstrated effective diagnosis capabilities for spine fractures and lesions; however, intricate medical imaging challenges require further research and innovative approaches. In May 16, 2024, the Radiological Society of North America launched a public competition on Kaggle for the development of Artificial Intelligence methods that assist in the classification and detection of degenerative spine conditions. This challenge offers an MR image dataset consisting of five lumbar spine degenerative conditions, categorized into three degrees of degeneration severity. In this work, we propose a multi-view hybrid quantum deep learning model and its classical counterpart based on a pretrained ResNet18 architecture. The data is split into sagittal T1, axial T2, and sagittal T2/STIR view samples, which are used to train the unimodal models of both approaches for the classes of normal/mild, moderate, and severe. A quantum circuit that functions as a convolutional and pooling layer is introduced as a final feature extraction layer, forming the hybrid quantum model. This hybrid quantum model approach demonstrates superior accuracy, precision, recall, and F1 scores of 0.8433, 0.8749, 0.8433, and 0.8550, respectively, outperforming its classical counterpart across all MR image perspectives. The proposed multi-view hybrid

L. Daniel Alejandro · M. Oscar (✉)
Instituto Politécnico Nacional—CITEDI, Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

L. Daniel Alejandro
e-mail: dlopez@citedi.mx

C. Oscar
Instituto Tecnológico de Tijuana, Tijuana, Baja California, Mexico
e-mail: ocastillo@tectijuana.mx

L. Miguel
ITJ Labs, Tijuana, Baja California, Mexico
e-mail: milopez@citedi.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_2

quantum learning approach enhances classification performance, reduces overfitting and loss, and shows promising capabilities in fine feature extraction and in addressing challenges in Computer-Aided Diagnosis (CAD) for the classification of spine degenerative conditions.

Keywords Spine lesion · Spine degenerative condition · Computer aided diagnosis · Deep learning · CNNs · Multi-view learning · Quantum machine learning · RSNA

1 Introduction

Spine lesions affect millions of patients per year in North America [1], ranging from conditions such as osteoporosis, fractures, and degenerative diseases, proving that spine injuries are an important problem to address. Previously in 2022, the Radiological Society of North America (RSNA) collaborated with the American Society of Neuroradiology (ASNR) and created the largest publicly available expert-labeled dataset of cervical spine fracture CT images for Artificial Intelligence (AI) research, which featured in the Kaggle website as the "RSNA 2022 AI Challenge". This dataset hosted a machine learning competition to help develop algorithms that assist in the detection of cervical spine fractures [1]. In 2024, the competition held by the RSNA and ASNR focused on the classification of five lumbar spine degenerative conditions defined as: Left neural foraminal narrowing, right neural foraminal narrowing, left subarticular stenosis, right subarticular stenosis, and spinal canal stenosis. Each imaging study presents severity scores (normal/mild, moderate, severe) for each of the five conditions across the intervertebral disc levels L1/L2, L2/L3, L3/L4, L4/L5, and L5/S1 [2]. Recent advancements in the application of AI for medical image analysis have enabled intelligent computer vision algorithms to assist medical experts in diagnosis, achieving results comparable to humans [3–5]. For spine fracture and lesion diagnosis, Deep Learning (DL) models have shown promising results in spine lesion image classification [6–9]. However, despite these advancements, DL methods still face limitations when addressing intricate medical problems. To overcome these challenges, innovative methodologies such as Multi-View (MV) learning have gained attention in recent years due to their representation capabilities for problems that require the use of different data types and image perspectives [10], as they combine different features and data sources for generalization [11]. MV Convolutional Neural Networks (CNNs) aim to work with multiple feature sets of the target, which is advantageous for 3D shape recognition problems, multivariate data, and multi-feature aggregation [10]. Furthermore, MV approaches present improvements for medical problems, as seen in [12], where time, frequency, and spatial information from electroencephalography samples are extracted and used for model training. Similarly, MV methods can also improve medical imaging problems as seen in [13], where a multi-model, multi-size, and multi-view approach for MR brain image segmentation is presented using transaxial, coronal, and sagittal image slices. Another

emerging technology that shows potential when working together with machine and deep learning models is quantum computing, particularly quantum machine learning, as it has shown promising results in complex feature abstraction [14, 15]. Although hybrid quantum machine learning approaches have been successfully applied to various medical imaging problems [16–18], there is still a lack of documented research exploring their use in spine fracture and lesion diagnosis.

Because of this, in this work, a solution for the RSNA 2024 lumbar spine degenerative classification competition is proposed, which focuses on a multi-view hybrid quantum deep learning model, as well as a classical counterpart based on the pre-trained ResNet18 architecture. The training data is split into sagittal T1, axial T2, and sagittal T2/STIR Magnetic Resonance (MR) views to train each of the unimodal models and predict the spine degeneration classes of normal/mild, moderate, and severe. The hybrid quantum approach achieves accuracy, precision, recall, and F1 score of 0.8433, 0.8749, 0.8433 and 0.8550, respectively, outperforming its classical counterpart on all MR image perspectives. This is achieved by a quantum circuit that functions as a layer inside the model's architecture and improves model performance, while also reducing overfitting and loss.

This book chapter is divided into the following sections: The spine degenerative condition problem is defined in Sect. 1, while some of the fundamentals used for this work are shown in Sect. 2. After this, the followed methodology is presented in Sect. 3, and the carried out experiments, as well as a brief discussion in Sect. 4. The conclusions and future work are located in Sect. 5, where the main contributions of this work and potential improvements are mentioned.

2 Fundamentals

2.1 Quantum Computing

Similar to classical computing, quantum computing uses a basic unit of information defined as a *qubit*, and its difference from the classical bit is that it can represent any two basis states at the same time as stated in the quantum physics fundamental principle of **superposition**. Qubits are commonly defined in the $z-$basis states where the basis states $|0\rangle$ and $|1\rangle$ represent the classical "0" and "1" bits [19]. To represent the two states that a qubit can be in between, Eq. 1 shows a combination of two terms that work with the variables α and β, which are complex numbers that represent the collapse probability of the state, this equation is written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where $|\psi\rangle$ represents a qubit, α, β the probability magnitudes, and $|0\rangle$, $|1\rangle$ the basis states that it can collapse to [19].

This expression must fulfill the normalization condition $|\alpha|^2 + |\beta|^2 = 1$, where all possible outcomes add up to 1. Similarly to classical computing, the combination of multiple qubits give rise to a **composite quantum system**, which is used to describe the interaction between qubits of the quantum system, as well as with other quantum systems [20]. A quantum system represented by Ψ is written as the tensor product for every qubit that it encompasses, written as

$$|\Psi\rangle = |\psi_1\rangle \otimes ... \otimes |\psi_n\rangle \,. \tag{2}$$

To facilitate representation in quantum circuits, the quantum system Ψ can be represented in base-2 notation as $|\Psi\rangle = c_0 \,|00\ldots0\rangle + c_1 \,|00\ldots1\rangle + \cdots + c_{2^n-1} = \sum_{i=0}^{2^n-1} c_i \,|i\rangle_2$, where the probability magnitude is represented by c_i, and the different possible states as $|i\rangle$ [19].

To manipulate qubits in a composite quantum system, **quantum gates** are implemented to transform their states. Analogous to logical classical gates, quantum gates allow to modify qubit states following the quantum physics principles of unitary and superposition. A quantum gate is normally expressed as U and must fulfill the condition $UU^\dagger = I$, meaning that the operation of the quantum gate and its conjugate transpose must result in the identity matrix I. The implementation of a quantum gate over a quantum system is written as

$$U\,|\Psi\rangle = U \sum_{i=0}^{2^n-1} \alpha_i \,|i\rangle \,, \tag{3}$$

where U represents the quantum gate, and $\sum_{i=0}^{2^n-1}$ the sum of the probability magnitudes α for the possible states $|i\rangle$ [20].

In quantum mechanics, quantum systems are described by wavefunctions that can be represented in superposition, and when a **measurement** is performed, it collapses to one of the possible outcomes. In quantum computing, after quantum processing in a system is complete, a measurement allows information retrieval from the quantum state. Typically, a projective measurement is done along the z-axis, collapsing the system into one of the probability amplitudes of a particular state that comprise the quantum system [20]. The quantum measurement can be represented using **spectral decomposition** of an observable, by establishing a quantum observable as M, the measurement can be expressed as:

$$M = \sum_m m\, P_m, \tag{4}$$

where M represents a quantum observable, m the possible outcomes, and P_m the projectors onto the corresponding eigenstates.

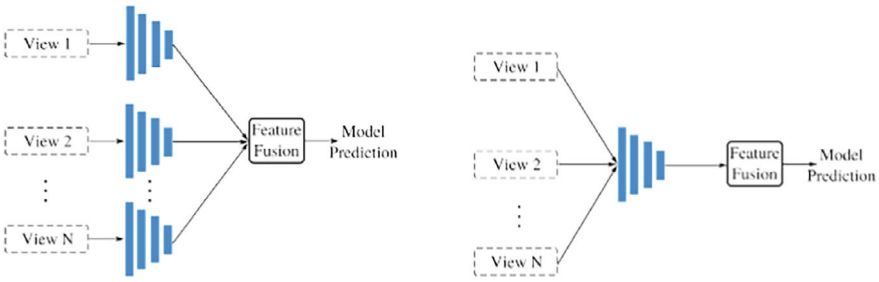
2.2 Multi-view Deep Learning Models

The MV methodology has demonstrated representation improvements when working with multivariate data, multifeature data, 3D images, as well as 2D images with multiple perspectives, such as MR images [10]. The main MV approaches are: One-view-one-net, and multi-view-one-net, as shown in Fig. 1.

The one-view-one-net method uses one CNN architecture for every provided view and extracts features for each of them, combining them at subsequent sections of the network. On the other hand, multi-view-one-net methods feed all views into the same CNN architecture, and then the feature fusion and model prediction steps follow it. The multi-view-one-net approach normally implements 3D convolutional, max-pooling and fully-connected layers to extract the final features [10].

3 Methodology

In this section, we detail the methodology employed to address the classification and detection of degenerative spine conditions using advanced machine learning techniques. Our approach integrates classical deep learning models with emerging quantum computing methods to enhance the accuracy and efficiency of medical image analysis. The subsequent sections describe the dataset, the multi-view learning model, the hybrid quantum model, and the classification metrics employed in our study.



(a) One-view-one-network method. (b) Multi-view-one-network method.

Fig. 1 Main multi-view deep learning approaches. Adapted from [10]

3.1 Dataset

The dataset used corresponds to the competition held by the Radiological Society of North America (RSNA) and the American Society of Neuroradiology (ASNR), which aims to explore the aid of artificial intelligence methods for classification and detection of degenerative spine conditions using lumbar spine Magnetic Resonance (MR) images. The dataset is comprised of 48,000 spine condition samples divided into five conditions with different levels of degenerative severity [2]. The degenerative spine conditions considered in the dataset are: Left Neural Foraminal Narrowing, right neural foraminal narrowing, left subarticular stenosis, right subarticular stenosis, and spinal canal stenosis. Every imaging study in the dataset is provided with a severity score of normal/mild, moderate, or severe for each of the conditions across the intervertebral disc levels L1/L2, L2/L3, L3/L4, L4/L5, and L5/S1 [2]. Data distribution across all five conditions and their severities is depicted in Fig. 2, where it is observed that the "normal/mild" class is the majority.

Every study has been curated and labeled by medical experts with Regions of Interest (ROI) for each intervertebral disc level. In this work, the provided ROI data points are used as reference to compute the bounding box for each data point and crop the disc level with its corresponding degenerative condition, as shown in Fig. 3. All spine degenerative condition samples are split into the three different MR image perspectives: Sagittal T1, Axial T2, and Sagittal T2/STIR. This is done to train the proposed model architecture with every perspective at different instances. The final set of images is resized to (224, 224), with 19,694 for Sagittal T1 samples, 19,215 for Axial T2 samples, and 9,748 for Sagittal T2/STIR samples. A batch of samples for each MR perspective and their corresponding class can be seen in Fig. 4.

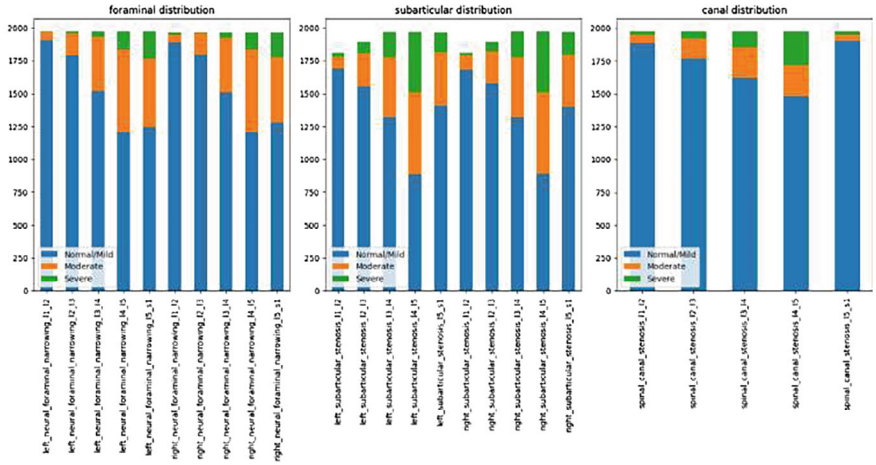
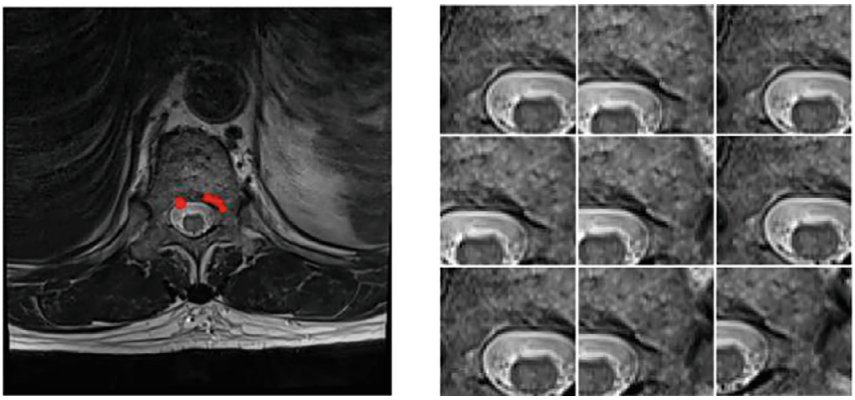


Fig. 2 Data distribution for the RSNA 2024 challenge dataset



(a) Spine degenerative condition ROI definition.

(b) Cropped ROI for spine degenerative condition.

Fig. 3 Spine ROI cropping process

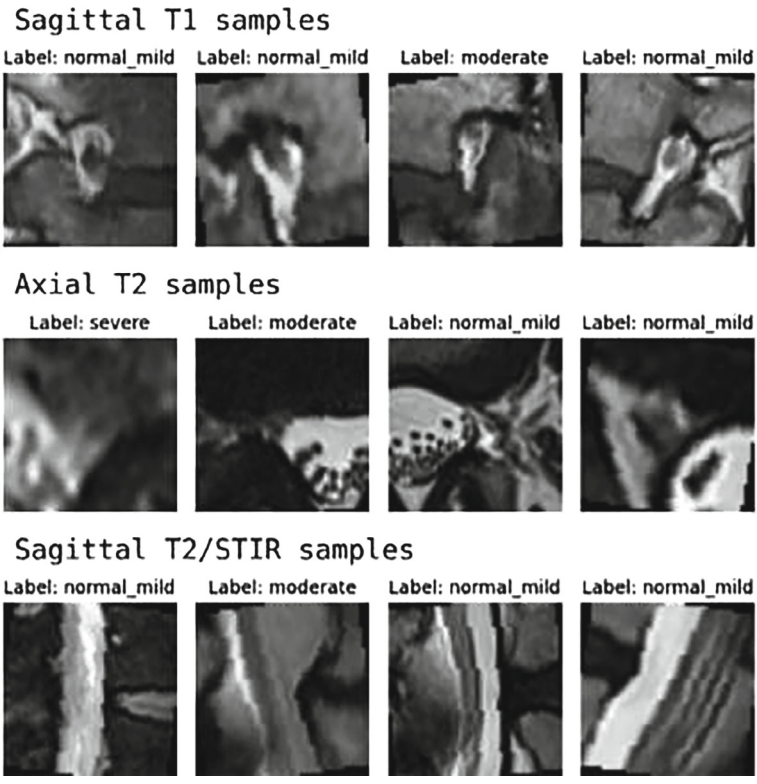


Fig. 4 Cropped intervertebral disc levels for every MR perspective and their spine condition label

3.2 *Multi-view Deep Learning Model*

The MV methodology has been widely used in medical problems, enabling different types of medical data to be fed to deep learning models to improve feature extraction and overall diagnosis performance [12, 13, 21–23]. In this work, the ResNet18 [24] model is used as the main architecture and repeated three times, formulating an MV one-view-one-net method, where each network trains with its corresponding MR image view as shown in the diagram of Fig. 5. The defined MV approach uses the three MR image perspectives of Sagittal T1, Axial T2, and Sagittal T2/STIR as input for every network of the ResNet18 architecture, extracting features for each set of images. After encoding and generating the feature maps, the label and image view are used as conditionals for the fusion module, enabling the corresponding model to provide a classification prediction based on the image view.

3.3 *Hybrid Quantum Multi-view Model*

The hybrid quantum model is proposed by integrating the proposed MV classic approach with a proposed quantum circuit that serves as a final feature extraction layer before to the network's classifier. The proposed quantum circuit is formulated using the PennyLane library [25]. The circuit can be separated into three main sections:

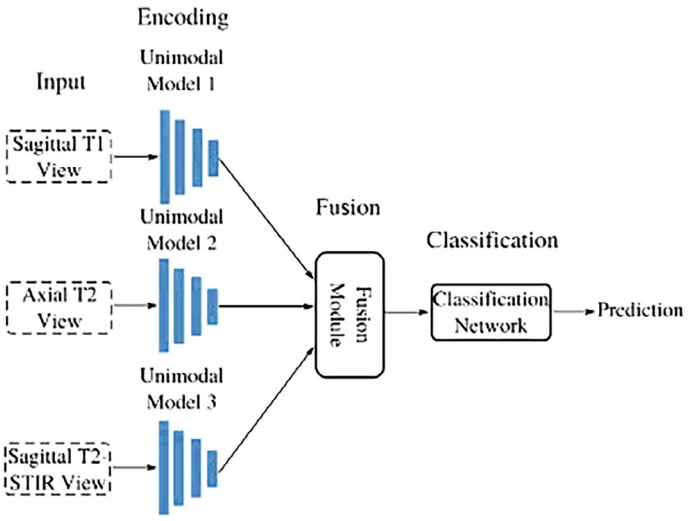


Fig. 5 Multi-view classical approach for RSNA 2024 challenge

- **Encoding**: An AngleEmbedding gate encodes the classical input from the deep learning model into the quantum state through parameterized rotations, preparing it for processing by subsequent quantum gates. This operation uses $RY(\theta)$ rotation gates, where θ represents the rotation angle derived from classical data parameters.
- **Quantum processing**: Once classical data has been encoded to the quantum state, additional quantum gates are utilized for data processing and feature extraction. In this case, a series of RY and CNOT gates form a quantum block that functions similarly to a classical convolutional layer. The CNOT gates entangle data between qubits and change their state according to the state of their corresponding control qubits. This facilitates correlation among input data features and enhances finer feature extraction. In addition, a quantum pooling layer is used to average out feature maps generated by the previous quantum gates, employing Hadamard gates, which produce superpositions to the applied qubits, alongside CNOT gates.
- **Decoding**: After quantum processing and feature extraction, the data must be converted back to the classical state. Measurement gates along the $z-$axis at the end of the quantum circuit are used for every qubit. This process collapses the data from the qubits back to its classical state so that it can be further processed and evaluated.

The proposed quantum circuit has 4 qubits to accommodate for the size dimensions of the final layers in the ResNet18 architecture, as shown in Fig. 6. The quantum circuit presents 8 trainable parameters updated after every training epoch, which adjust the states of the parameters affected by the implemented quantum gates. The quantum convolutional and pooling layers improve the generalization of small details present in spine degenerative conditions. After defining the quantum circuit for the hybrid quantum architecture, it is integrated as a trainable layer inside the classical ResNet18 model at the very end of the feature extraction phase. The resulting hybrid quantum multi-view approach is very similar to its classical counterpart, however, instead of using the unimodal classical model for a particular MR image view, its hybrid quantum version is used instead.

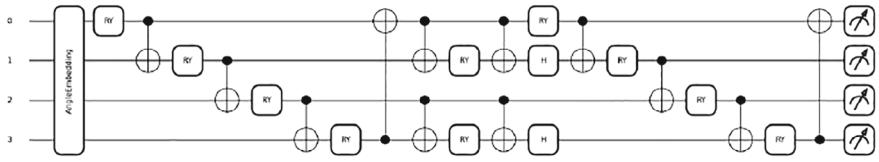


Fig. 6 Proposed quantum circuit

3.4 Classification Metrics

The proposed MV approach for the RSNA 2024 challenge is evaluated differently to how regular datasets are evaluated, although the main train set includes labels for every sample, the test set does not, and the RSNA expects model probabilities for each of the degenerative spine conditions. Because of this, the training set is evaluated using the main classification metrics of Accuracy (ACC), Recall (R), Precision (P), and F1 Score (F1), computed by using the predicted values for every classified sample. Model predictions classified as positive and belonging to the predicted class are represented as True Positive (TP) values, while correctly predicted negative samples are True Negative (TN) values. On the other hand, when the model mistakenly predicts samples that do not belong to the reviewed class as positive and negative they represent False Positive (FP) and False Negative (FN) values respectively [26]. Computation of the classification metrics is expressed as:

$$\begin{aligned} P &= \frac{TP}{TP + FP}, \\ R &= \frac{TP}{TP + FN}, \\ ACC &= \frac{TP + TN}{TP + TN + FP + FN}, \\ F1 &= 2\left(\frac{P \times R}{P + R}\right). \end{aligned} \tag{5}$$

Regarding test set performance evaluation, the MV model performance is evaluated using the Gradient-weighted Class Activation Mapping (Grad-CAM) [27] tool to analyze the model's behavior. The Grad-CAM generates a visual heatmap that highlights the region of the image that contributed the most to the model's decision. By identifying where the model is focusing when making predictions, an assessment on the relevant input information can be made. This information aids in unlabeled data performance evaluation as it enables the visualization of the most relevant sample information for the model and it can be further used as a diagnosis tool for physician's interpretation.

Finally, as stated in the RSNA 2024 competition, the proposal's predicted probabilities are used to evaluate model performance on test set samples. In this case, probability distribution bar plots are used to observe decision making and outlook on the model's interpretation of the test set.

4 Experiments and Results

The experiments are conducted on an Ubuntu 22.04 system eqquiped with CUDA toolkit 12.5 using the Pytorch framework with torch version 2.0.1. Quantum circuit simulations are performed using the PennyLane 0.37 library. The computing hardware includes and AMD Ryzen Threadripper 1920X CPU and an NVIDIA RTX 3080. The ResNet18 architecture is pretrained on the ImageNet1K_V2 dataset [28]. The crossentropy loss function is weighted according to the number of degenerative spine condition samples, adjusting the weights and increasing relevance to the minority "severe" class. The optimizer used is Adam [29] with a learning rate of 1×10^{-4}, a learning rate scheduler applied every 2 steps, with a batch size of 32 for 25 epochs and a patience of 5.

Overview of Experiments: A series of four primary experiments were conducted to evaluate the effectiveness of the multi-view (MV) classical and hybrid quantum models across various MR image perspectives. Each experiment focused on different aspects of the model performance, including accuracy, loss, and other classification metrics. Advanced diagnostic tools such as Grad-CAM were employed to analyze model behavior on the test set, providing insights into the model's decision-making processes.

The first experiment aims to evaluate the performance of the MV classical model across various MR image perspectives to identify degenerative spine conditions. First, the results of the MV model show an accuracy of 0.61 for the train and validation sets, as well as a loss of 0.96. These results suggest that although there are plenty of Sagittal T1 samples, the classical MV model still struggles to identify degenerative spine conditions in this MR image perspective. The behavior observed on the accuracy and loss curves of Fig. 7 shows favorable results, but there are difficulties overall considering the number of samples used for training. The rest of the classification metrics such as recall, precision, and F1 score are shown in the curves of Fig. 8, where it can be seen that although the recall stays around 0.61 similarly to the training accuracy, the model precision increases considerably to 0.73, and in consequence,

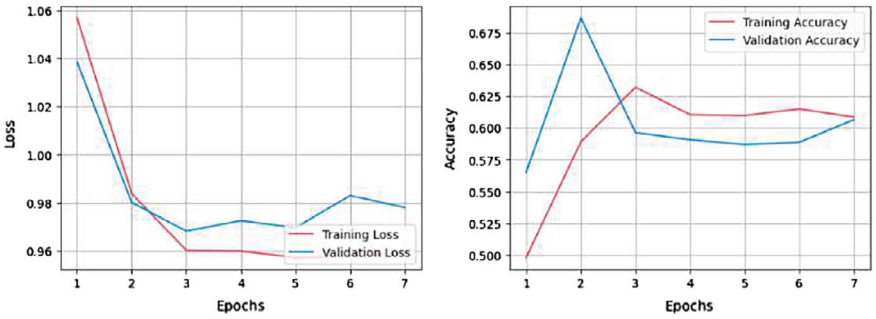


Fig. 7 Accuracy and loss curves for classical model on Sagittal T1 images

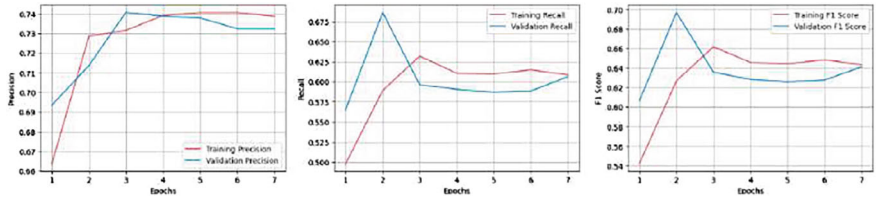


Fig. 8 Recall, precision, and F1 score curves for classical model on Sagittal T1 images

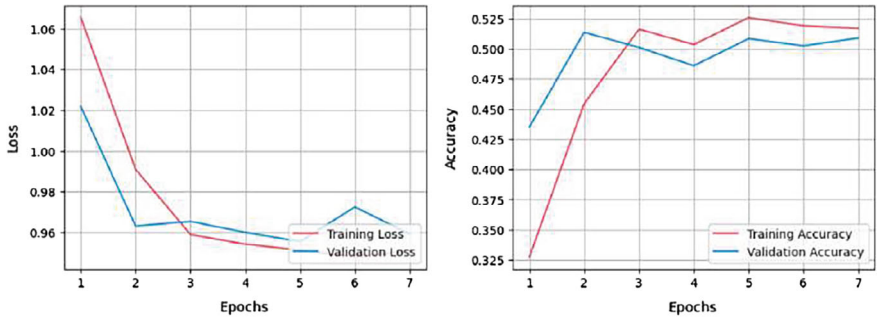


Fig. 9 Accuracy and loss curves for classical model on Axial T2 images

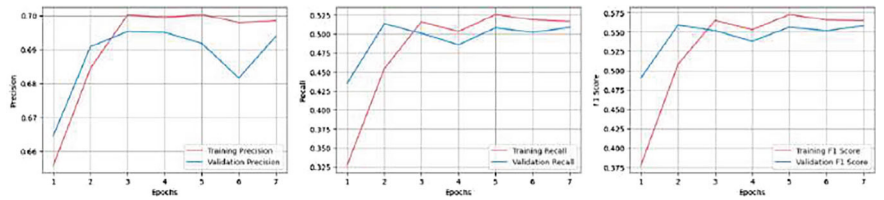


Fig. 10 Recall, precision, and F1 score curves for classical model on Axial T2 images

the achieved F1 score is 0.64. These results show that although it is able to correctly predict spine conditions on this MR perspective, there are still improvements to be made.

The results of the MV model for the Axial T2 perspective are lower, reaching an accuracy of around 0.51 for the training and validation sets as observed in Fig. 9. This is mostly due to the difficulties in identifying the different intervertebral disc levels using this set of slices. The results shown in Fig. 10 demonstrate a slight increase in performance for the precision attained by achieving 0.70, but still attain subpar results for the recall, and F1 score of 0.52.

The first experiment concludes with the training the MV model with the Sagittal T2/STIR MR perspective, where the model achieves the highest results as seen in Fig. 11. The accuracy obtained is 0.83 and 0.82 for the training and validation sets respectively, showing that this perspective is the best one to generalize the problem and distinguish different grades of spine degeneration. The results for the rest of

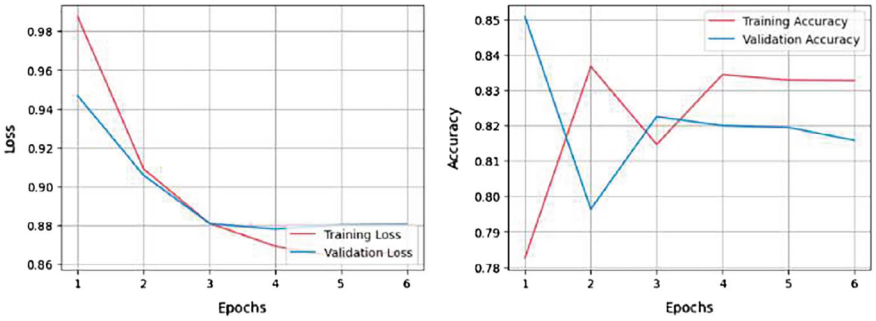


Fig. 11 Accuracy and loss curves for classical model on Sagittal T2/STIR images

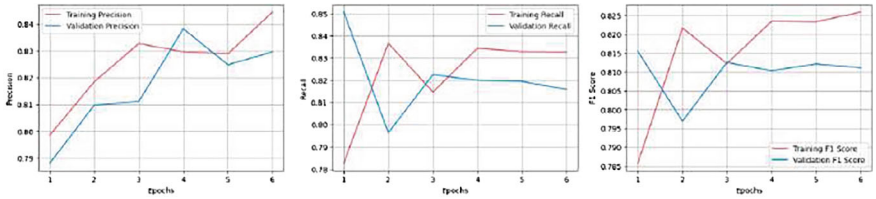


Fig. 12 Recall, precision, and F1 score curves for classical model on Sagittal T2/STIR images

the classification metrics also achieve the highest values as shown in Fig. 12, where recall, precision, and F1 score attain 0.83, 0.83, and 0.82 respectively, demonstrating that the model is able to identify both positive and negative samples between the different intervertebral disc levels and their levels of spine degeneration.

The second experiment aims to evaluate the test set using the Grad-CAM layer set at the final layer of the model, and as seen in Fig. 13, the Grad-CAM heatmaps generated for each of the analyzed test set samples focus on different regions of the spine; however, the results still show some inconsistency that can be attributed to different samples and intervertebral disc levels being analyzed, or a lack of focus that indicates an overfitting and instability problem. Nevertheless, considering the differences and the data distribution along the training set, the model demonstrates favorable generalization of the problem, while avoiding biases towards the majority classes.

The classical MV model is evaluated on the predicted probabilities of each spine degenerative condition for each test set sample. Although no labels are provided for the test set, observing the probability distribution bar plots of Fig. 14, model behavior can be evaluated. The predicted probabilities for the "Severe" class show that the model is certain that most samples of the test set do not belong to this class, which is supported by the training data distribution, where it is the minority class. The probabilities for the "Moderate" class showcase that the model is somewhat certain that most samples could be either from this class or the majority class "Normal/Mild", as most probability values range between [0.3, 0.4]. Finally, for the "Normal/Mild"

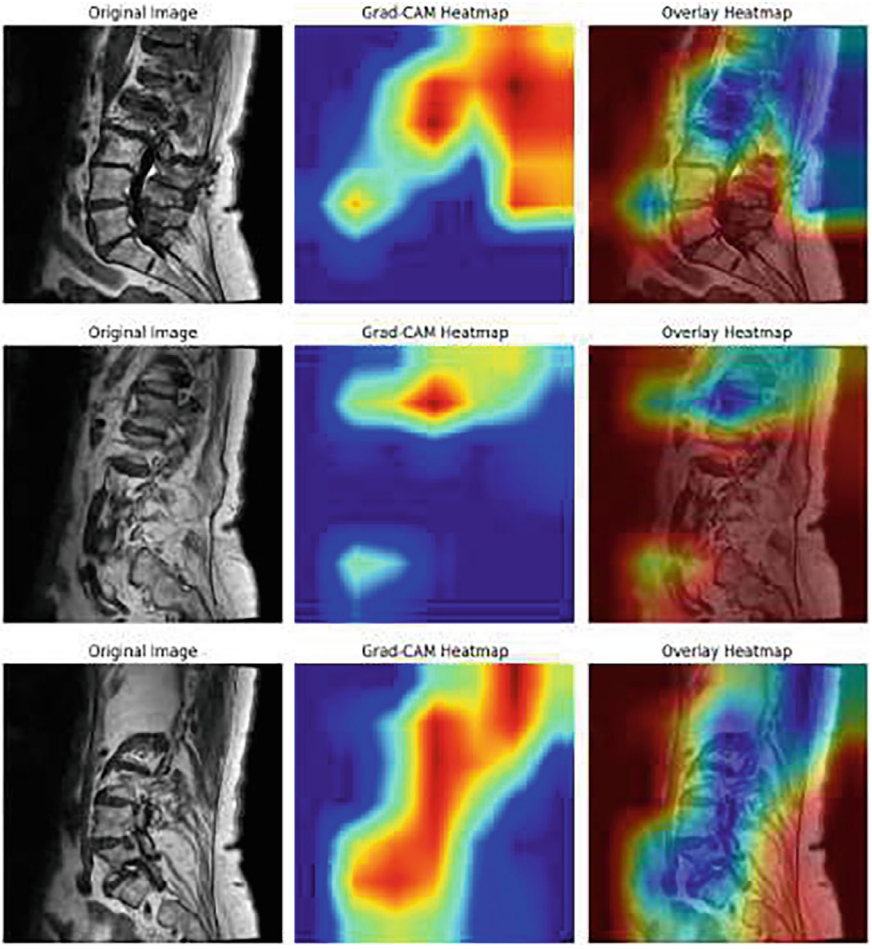


Fig. 13 Classical model Grad-CAM results for the test set

class, there is a gap for the predicted probabilities, where the model is indecisive if the spine condition is "Normal/Mild". However, this behavior shows that the model generalizes well but still presents some difficulties with the minority class, as well as difficult to identify samples.

The third experiment focuses on the training of the hybrid quantum MV model for every MR image perspective to evaluate the enhancement in model performance due to quantum layer feature extraction capabilities. The accuracy and loss results of the hybrid quantum model shown in Fig. 15 for the Sagittal T1 MR perspective improve when compared to its classical counterpart, reaching a lower loss of 0.82 and an accuracy of 0.66 for both the training and validation sets, showing the quantum layer feature extraction capabilities for this perspective. The results for the classification

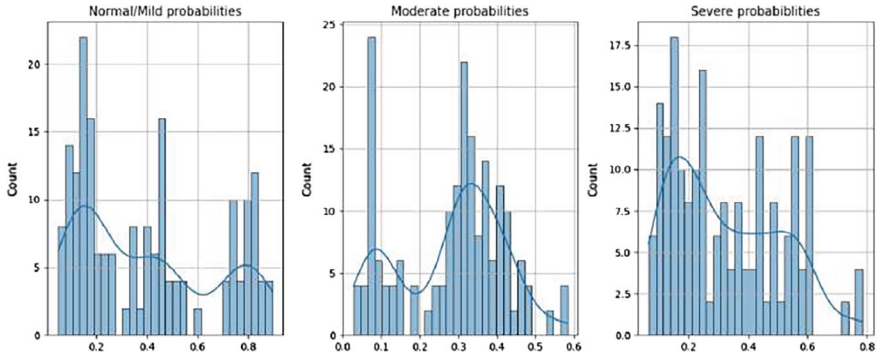


Fig. 14 Classical model predicted probabilities for the test set

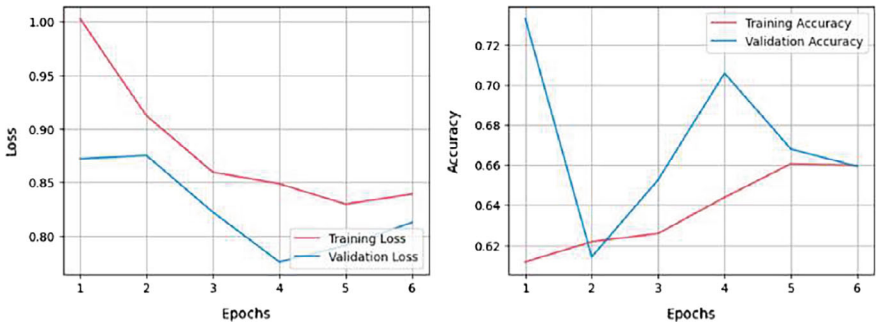


Fig. 15 Accuracy and loss curves for hybrid quantum model on Sagittal T1 images

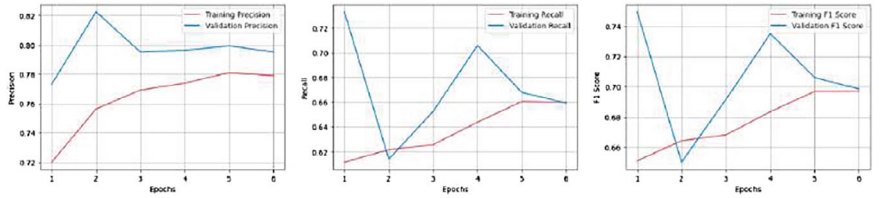


Fig. 16 Recall, precision, and F1 score curves for hybrid quantum model on Sagittal T1 images

metrics of precision, recall, and F1 score for the hybrid quantum model suggest the same, increasing overall performance by 10% as seen in the curves of Fig. 16, where the recall, precision, and F1 score attained are 0.66, 0.78, and 0.70 respectively.

Although the Axial T2 MR perspective presents more difficulties in identifying intervertebral spaces, the hybrid quantum model is able to improve accuracy by 10% when compared to its classical counterpart as seen in Fig. 17, reaching an accuracy of 0.62. The same can be said regarding the remaining classification metrics of precision, recall, and F1 score shown in the curves of Fig. 18, where the model is

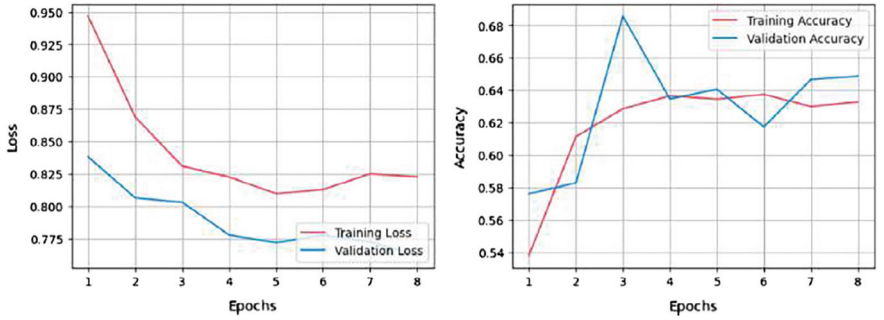


Fig. 17 Accuracy and loss curves for hybrid quantum model on Axial T2 images

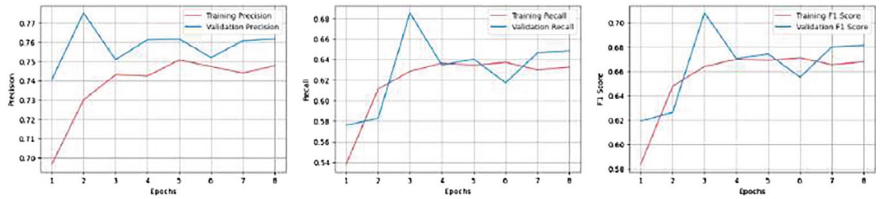


Fig. 18 Recall, precision, and F1 score curves for hybrid quantum model on Axial T2 images

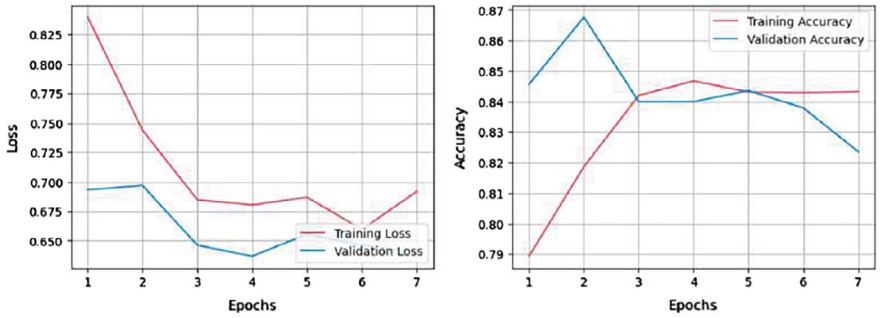


Fig. 19 Accuracy and loss curves for hybrid quantum model on Sagittal T2 images

able to generalize better the spine degenerative conditions and attain 0.75, 0.64, and 0.67 respectively, an increase of 10% from what the classical model achieved.

The third experiment is concluded by training the hybrid quantum MV model on the Sagittal T2/STIR MR perspective, where the model achieves 0.84 of accuracy, a 3% increase compared to its classical counterpart as seen in Fig. 20. Furthermore, the results for the remaining classification metrics of precision, recall and F1 score outperform the classical model as well, achieving 0.86, 0.82, and 0.83 respectively, an increase of 3% from what the classical model attained. Despite the fact that the Sagittal T2/STIR perspective is the easiest one to differentiate intervertebral disc levels, a performance improvement is still observed as demonstrated in Fig. 20.

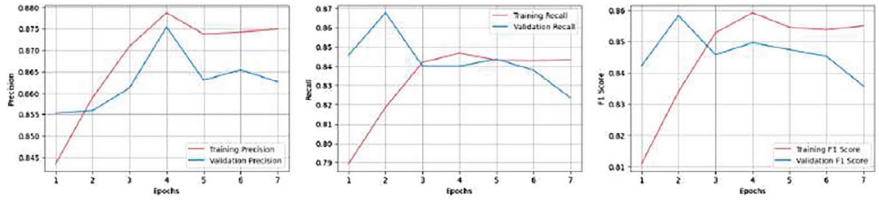


Fig. 20 Recall, precision, and F1 score curves for hybrid quantum model on Sagittal T2/STIR images

The fourth experiment centers on evaluating the hybrid quantum MV model performance on the unlabeled test set. As stated previously, the Grad-CAM tool is used as part of the last layer of the hybrid quantum model to analyze its behavior and decision making. The generated heatmap seen in Fig. 21 shows the regions that the hybrid quantum model focuses on when making predictions, where although it highlights sections of the spine, it also showcases instability and difficulties in identifying the spine on the image.

To conclude the hybrid quantum evaluation and the fourth experiment, the predicted probabilities for the test set are plotted as seen in Fig. 22. Similarly to its classical counterpart, the hybrid quantum model is certain that most of the samples in the test set do not present severe spine degeneration. Additionally, most predictions are towards the "Moderate" class, where the number of probability predictions in the range of [0.3, 0.5] is relatively high when compared to the other classes. However, the hybrid quantum model still presents some difficulties when differentiating between the "Normal/Mild" and "Moderate" classes, as seen in the probability predictions of the "Normal/Mild" class, where there is a considerable amount of predictions in the range of [0.4, 0.8] due to it being the majority class. Even considering the difficulties when generating an appropriate heatmap for the test samples, as well as a degree of uncertainty on the probability predictions, the hybrid quantum model showcases cautiousness when making a decision, a key factor when working with classes defined by very fine details.

After training and evaluating both MV method proposals, their classification results for the training set are compiled in Table 1, where performance differences are more apparent. As stated previously, the hybrid quantum model outperforms its classical counterpart on all metrics for all MR image perspectives on both the training and validation sets. These results demonstrate the generalization improvements attained by using the quantum layer as a final feature extractor for the architecture, refining small feature extraction and mitigating loss increases as observed in the curves of Figs. 15, 17, and 19. Additionally, the hybrid quantum model is able to improve accuracy, precision, recall, and F1 score on the training and validation sets for the Axial T2 perspective by an average of 10%, which helps make the model's assessment considerably more reliable than before; the same can be said for the remaining MR image perspectives, increasing performance for all classification metrics for the Sagittal T1 and Sagittal T2/STIR perspectives on an average of 5% and 3% respectively.

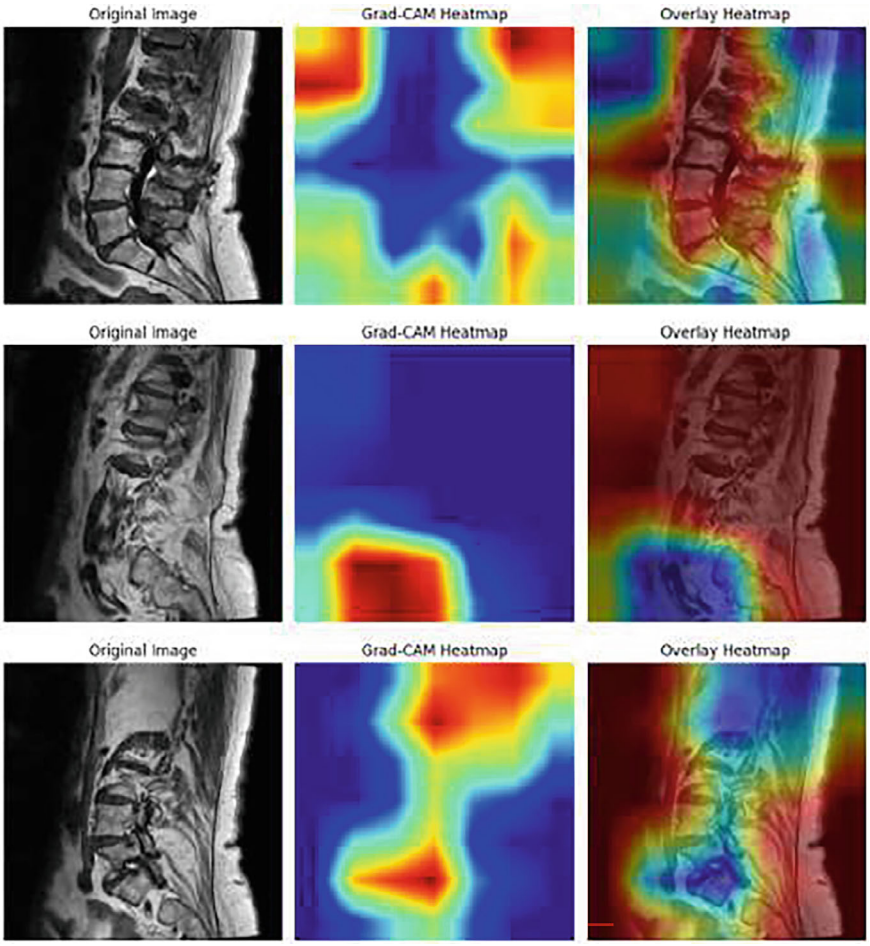


Fig. 21 Hybrid quantum model Grad-CAM results for the test set

5 Conclusions and Future Work

Spine degenerative conditions affect a great number of people worldwide, and its early diagnosis is fundamental to preventing spine lesions that compromise their health. Because of this, medical imaging tools such as MRIs provide insightful information across different perspectives, enabling detailed diagnosis for lumbar spine vertebrae, discs and nerves. This work addresses on tackling the challenge set by the RSNA in 2024 on Kaggle, focusing on the diagnosis of three main spine degenerative conditions across the intervertebral disc levels L1/L2, L2/L3, L3/L4, and L5/S1.

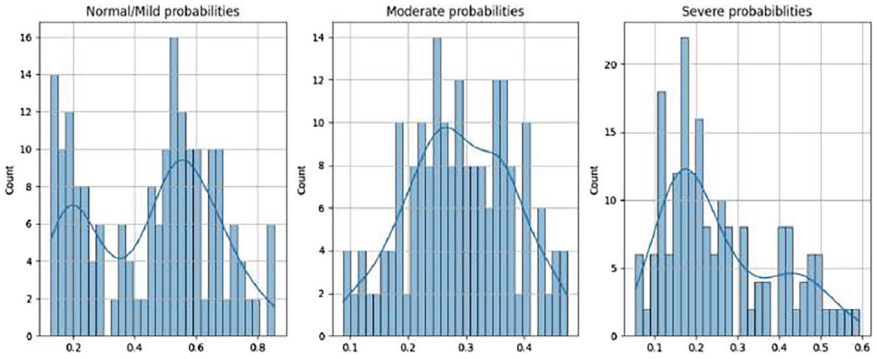


Fig. 22 Hybrid quantum model predicted probabilities for the test set

Table 1 Multi-view classification model comparison. Best results depending on model view in bold

Model type	Model-view	Train				Validation			
		ACC	P	R	F1	ACC	P	R	F1
Classical	SagittalT1	0.6086	0.7388	0.6086	0.6435	0.6065	0.7324	0.6065	0.6415
	AxialT2	0.5167	0.6986	0.5167	0.5650	0.5087	0.6940	0.5087	0.5583
	SagittalT2/STIR	0.8328	0.8445	0.8328	0.8259	0.8159	0.8297	0.8159	0.8111
Hybrid quantum	SagittalT1	**0.6597**	**0.7791**	**0.6597**	**0.6970**	**0.6593**	**0.7952**	**0.6593**	**0.6986**
	AxialT2	**0.6328**	**0.7479**	**0.6328**	**0.6678**	**0.6487**	**0.7618**	**0.6487**	**0.6814**
	SagittalT2/STIR	**0.8433**	**0.8749**	**0.8433**	**0.8550**	**0.8236**	**0.8626**	**0.8236**	**0.8358**

Our approach includes the implementation of a multi-view hybrid quantum deep learning model and its classical counterpart based on the pretrained ResNet18 architecture. The training samples are split into three groups for the MR perspectives of sagittal T1, axial T2, and sagittal T2/STIR, each representing different spine degeneration degrees: normal/mild, moderate, and severe. The multi-view methodology establishes three unimodal models trained on a specific MR image view, which are then combined through study and series labels for model prediction.

To formulate the hybrid quantum unimodal models, a quantum circuit that implements two quantum convolutions and a quantum pooling layer is used as a final feature extraction layer before the architecture's classifier. Four main experiments are carried out to evaluate performance of the proposed multi-view models for the train, validation and test sets, where the results for the training and validation sets show an improvement of 10%, 5%, and 3% in favor of the hybrid quantum approach for the MR views of axial T2, sagittal T1, and sagittal T2/STIR respectively.

Both the classical and hybrid quantum model achieve the highest results on the sagittal T2/STIR view, with the classical model attaining an accuracy, precision, recall

and F1 score of 0.8328, 0.8445, 0.8328, and 0.8259; while the hybrid quantum model improves these results by reaching 0.8433, 0.8749, 0.8433, and 0.8550, respectively. These training results showcase the capability of the proposed quantum layer to improve fine feature extraction, as well as loss and overfitting mitigation.

As indicated in the competition, the test set does not include labels so that the probabilities for spine degeneration condition are provided for evaluation. Because of this, the Grad-CAM tool and a probability distribution plot are used to support the test set results. The plotted heatmaps for both the classical and hybrid quantum model show that although both proposals make their decisions based on spine information from the MR image, there are still difficulties when identifying intervertebral disc spaces in several instances.

Regarding the probabilities distribution for the spine degeneration conditions, it is shown that even though both models present a degree of uncertainty for normal/mild and moderate samples, there is confidence that the severe minority class is almost non existent as only few cases are diagnosed as such. Furthermore, the hybrid quantum model shows a more continuous prediction distribution, as most probabilities fall under a range of [0.3, 0.4] for moderate samples, while also predicting the presence of normal/mild samples in some cases; decisions that the classic model is more uncertain about as seen in the predicted probabilities distribution gaps.

In conclusion, the proposed quantum circuit and hybrid quantum multi-view approach outperform the classical multi-view method in spine degenerative condition classification by 10%, 5%, and 3%, improving intricate feature extraction and reducing overfitting. Nevertheless, these results can be further improved in future work by considering different classification models, quantum circuits, data augmentation, as well as exploring more intricate approaches that make use of all the provided data in the RSNA 2024 challenge.

Acknowledgements The research leading to these results has received funding from the Consejo Nacional de Humanidades Ciencias y Tecnologias (CONAHCYT) through project CF-2023-I-108, and from Instituto Politécnico Nacional (IPN), under the Research Project SIP20240164.

References

1. Lin, H.M., Colak, E., Richards, T., Kitamura, F.C., Prevedello, L.M., Talbott, J., Ball, R.L., Gumeler, E., Yeom, K.W., Hamghalam, M., Simpson, A.L., Strika, J., Bulja, D., Angkurawaranon, S., Pérez-Lara, A., Gómez-Alonso, M.I., Jiménez, J.O., Peoples, J.J., Law, M., Dogan, H., Altinmakas, E., Youssef, A., Mahfouz, Y., Kalpathy-Cramer, J., Flanders, A.E.: The rsna cervical spine fracture ct dataset. Radiology: Artificial Intelligence **5** (2023). https://doi.org/10.1148/RYAI.230034
2. Richards, T., Talbott, J., Ball, R., Colak, E., Flanders, A., Kitamura, F., Mongan, J., Prevedello, L., Vazirabad., M.: RSNA 2024 Lumbar Spine Degenerative Classification. Kaggle (2024). https://kaggle.com/competitions/rsna-2024-lumbar-spine-degenerative-classification

3. Singh, D., Kumar, V., Vaishali, Kaur, M.: Classification of covid-19 patients from chest ct images using multi-objective differential evolution-based convolutional neural networks. European Journal of Clinical Microbiology and Infectious Diseases 39, 1379–1389 (2020). https://doi.org/10.1007/S10096-020-03901-Z/FIGURES/13
4. Reshi, A.A., Rustam, F., Mehmood, A., Alhossan, A., Alrabiah, Z., Ahmad, A., Alsuwailem, H., Choi, G.S.: An efficient cnn model for covid-19 disease detection based on x-ray image classification. Complexity **2021** (2021). https://doi.org/10.1155/2021/6621607
5. Brinker, T.J., Kiehl, L., Schmitt, M., Jutzi, T.B., Krieghoff-Henning, E.I., Krahl, D., Kutzner, H., Gholam, P., Haferkamp, S., Klode, J., Schadendorf, D., Hekler, A., Fröhling, S., Kather, J.N., Haggenmüller, S., Kalle, C., Heppt, M., Hilke, F., Ghoreschi, K., Tiemann, M., Wehkamp, U., Hauschild, A., Weichenthal, M., Utikal, J.S.: Deep learning approach to predict sentinel lymph node status directly from routine histology of primary melanoma tumours. European Journal of Cancer 154, 227–234 (2021). https://doi.org/10.1016/J.EJCA.2021.05.026
6. Murata, K., Endo, K., Aihara, T., Suzuki, H., Sawaji, Y., Matsuoka, Y., Nishimura, H., Takamatsu, T., Konishi, T., Maekawa, A., Yamauchi, H., Kanazawa, K., Endo, H., Tsuji, H., Inoue, S., Fukushima, N., Kikuchi, H., Sato, H., Yamamoto, K.: Artificial intelligence for the detection of vertebral fractures on plain spinal radiography. Scientific Reports **10** (2020). https://doi.org/10.1038/S41598-020-76866-W
7. Chen, H.Y., Hsu, B.W.Y., Yin, Y.K., Lin, F.H., Yang, T.H., Yang, R.S., Lee, C.K., Tseng, V.S.: Application of deep learning algorithm to detect and visualize vertebral fractures on plain frontal radiographs. PLoS ONE **16** (2021). https://doi.org/10.1371/JOURNAL.PONE.0245992
8. Yabu, A., Hoshino, M., Tabuchi, H., Takahashi, S., Masumoto, H., Akada, M., Morita, S., Maeno, T., Iwamae, M., Inose, H., Kato, T., Yoshii, T., Tsujio, T., Terai, H., Toyoda, H., Suzuki, A., Tamai, K., Ohyama, S., Hori, Y., Okawa, A., Nakamura, H.: Using artificial intelligence to diagnose fresh osteoporotic vertebral fractures on magnetic resonance images. Spine Journal 21, 1652–1658 (2021). https://doi.org/10.1016/J.SPINEE.2021.03.006
9. Rosenberg, G.S., Cina, A., Schiró, G.R., Giorgi, P.D., Gueorguiev, B., Alini, M., Varga, P., Galbusera, F., Gallazzi, E.: Artificial intelligence accurately detects traumatic thoracolumbar fractures on sagittal radiographs. Medicina 2022, Vol. 58, Page 998 58, 998 (2022). https://doi.org/10.3390/MEDICINA58080998
10. Yan, X., Hu, S., Mao, Y., Ye, Y., Yu, H.: Deep multi-view learning methods: A review. Neurocomputing 448, 106–129 (2021). https://doi.org/10.1016/J.NEUCOM.2021.03.090
11. Sun, S.: A survey of multi-view machine learning. Neural Computing and Applications 23, 2031–2038 (2013). https://doi.org/10.1007/S00521-013-1362-6/METRICS
12. Xu, J., Zheng, H., Wang, J., Li, D., Fang, X.: Recognition of eeg signal motor imagery intention based on deep multi-view feature learning. Sensors 2020, Vol. 20, Page 3496 20, 3496 (2020). https://doi.org/10.3390/S20123496
13. Wei, J., Xia, Y., Zhang, Y.: M3net: A multi-model, multi-size, and multi-view deep neural network for brain magnetic resonance image segmentation. Pattern Recognition 91, 366–378 (2019). https://doi.org/10.1016/J.PATCOG.2019.03.004
14. Maheshwari, D., Garcia-Zapirain, B., Sierra-Sosa, D.: Quantum machine learning applications in the biomedical domain: A systematic review. IEEE Access 10, 80463–80484 (2022). https://doi.org/10.1109/ACCESS.2022.3195044
15. Matondo-Mvula, N., Elleithy, K.: Advances in quantum medical image analysis using machine learning: Current status and future directions. Proceedings—2023 IEEE International Conference on Quantum Computing and Engineering, QCE 2023 1, 367–377 (2023). https://doi.org/10.1109/QCE57702.2023.00049
16. Bokhan, D., Mastiukova, A.S., Boev, A.S., Trubnikov, D.N., Fedorov, A.K.: Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. Frontiers in Physics 10, 1069985 (2022). https://doi.org/10.3389/FPHY.2022.1069985/BIBTEX
17. Houssein, E.H., Abohashima, Z., Elhoseny, M., Mohamed, W.M.: Hybrid quantum-classical convolutional neural network model for covid-19 prediction using chest x-ray images. Journal

of Computational Design and Engineering 9, 343–363 (2022). https://doi.org/10.1093/JCDE/QWAC003
18. Wei, L., Liu, H., Xu, J., Shi, L., Shan, Z., Zhao, B., Gao, Y.: Quantum machine learning in medical image analysis. Neurocomputing 525, 42–53 (2023). https://doi.org/10.1016/J.NEUCOM.2023.01.049
19. Ross, O.H.M.: A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. IEEE Access 8, 814–838 (2020). https://doi.org/10.1109/ACCESS.2019.2962155
20. Zheng, J., Gao, Q., Lü, J., Ogorzałek, M., Pan, Y., Lü, Y.: Design of a quantum convolutional neural network on quantum circuits. Journal of the Franklin Institute (2022). https://doi.org/10.1016/J.JFRANKLIN.2022.07.033
21. Wang, Y., Choi, E.J., Choi, Y., Zhang, H., Jin, G.Y., Ko, S.B.: Breast cancer classification in automated breast ultrasound using multiview convolutional neural network with transfer learning. Ultrasound in Medicine and Biology 46, 1119–1132 (2020). https://doi.org/10.1016/J.ULTRASMEDBIO.2020.01.001
22. Ma, J., Kou, W., Lin, M., Cho, C.C.M., Chiu, B.: Multiclass and multilabel classifications by consensus and complementarity-based multiview latent space projection. IEEE Transactions on Systems, Man, and Cybernetics: Systems 54, 1705–1718 (2024). https://doi.org/10.1109/TSMC.2023.3327925
23. Thenmoezhi, N., Perumal, B., Lakshmi, A.: Multi-view image fusion using ensemble deep learning algorithm for mri and ct images. ACM Transactions on Asian and Low-Resource Language Information Processing **23** (2024). https://doi.org/10.1145/3640811
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December, 770–778 (2015). https://doi.org/10.1109/CVPR.2016.90
25. Pennylane: Automatic differentiation of hybrid quantum-classical computations (2018). https://doi.org/10.48550/arxiv.1811.04968
26. Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview (2020). https://doi.org/10.48550/arxiv.2008.05756
27. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017). https://doi.org/10.1109/ICCV.2017.74
28. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
29. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings (2014). https://doi.org/10.48550/arxiv.1412.6980

Enhancing Nocturnal Blood Pressure Detection Through Interval Type-2 Fuzzy Classification Systems

Ivette Miramontes and Patricia Melin

Abstract In this work, we present the development and application of an Interval Type-2 Fuzzy System designed to classify nocturnal blood pressure, a crucial factor in determining the risk of heart disease. Traditional methods often face difficulties due to the inherent uncertainties and variations in blood pressure readings during sleep, which can affect classification accuracy. The proposed system utilizes the robustness of interval type-2 fuzzy logic to handle these uncertainties more effectively and accurately. This study details the classifier's architecture, including the selection and design of membership functions and the rule bases necessary for accurate classification. The system's performance is evaluated using a dataset collected from 20 patients, demonstrating significant improvements in classification accuracy and reliability when compared to Type-1 Fuzzy Systems. The results suggest that the Interval Type-2 Fuzzy System demonstrated high accuracy, with the trapezoidal membership functions achieving 95% classification precision and only one misclassification. Similarly, Gaussian membership functions yielded an 89% accuracy with two misclassifications. These findings indicate that the system is a promising tool for cardiologists in monitoring and managing nocturnal hypertension, offering improved prediction and prevention of heart diseases.

Keywords Interval Type-2 fuzzy system · Nocturnal blood pressure · Hypertension · Fuzzy logic

I. Miramontes (✉) · P. Melin
Tijuana Institute of Technology, Tijuana, BC, México
e-mail: cynthia.miramontes@tectijuana.edu.mx

P. Melin
e-mail: pmelin@tectijuana.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_3

1 Introduction

In today's digital era, where data is generated in massive volumes per minute, it is essential to have tools capable of efficiently managing and interpreting this information [1, 2]. However, in many cases, the available information can be incomplete, imprecise, or vague. To address this issue, Fuzzy Inference Systems (FIS) have been introduced, which are effective at handling imprecision in the data. Nevertheless, when the imprecision is significant, an alternative solution is Interval Type-2 Fuzzy Systems (IT2FSs). These systems have proven to be exceptional tools, especially in contexts where it is necessary to work with high levels of uncertainty and where the data may be affected by noise [3–5].

One of the most promising areas for the implementation of IT2FSs is medicine, where these systems are particularly useful in addressing diagnostic problems based on both clinical data and images. The ability of IT2FSs to handle high levels of uncertainty and tolerate noise in data is crucial in medical environments, where decisions often need to be made with incomplete or imprecise information. Various studies have validated the effectiveness of these systems in making critical decisions in medical care, highlighting their potential to transform modern clinical practice [6–10].

The main goal of this research is to evaluate the effectiveness of an IT2FS in classifying nocturnal blood pressure profiles. The study focuses on comparing the performance of two different membership functions—trapezoidal and Gaussian—by analyzing their impact on classification accuracy. Additionally, the system's parameters, such as lower lag and lower scale, were manually adjusted to optimize performance. The results highlight the system's potential in improving hypertension diagnosis, providing cardiologists with a more accurate tool for predicting nocturnal hypertension and, consequently, preventing related heart diseases.

The organization of this article is as follows: Sect. 2 provides a review of the literature, Sect. 3 details the employed methodology, Sect. 4 presents the analysis of the results, and Sect. 5 concludes with the findings of this study.

2 Literature Review

This section provides an overview of the fundamental concepts necessary to understand the problem

2.1 Blood Pressure and Hypertension

Blood pressure is defined as the force exerted by the blood against the walls of the arteries as it circulates. It is divided into systolic and diastolic pressure and is

measured in millimeters of mercury (mmHg) [11]. Systolic pressure corresponds to the highest value, recorded when the heart contracts and pumps blood, while diastolic pressure is the lowest value, recorded when the heart relaxes between beats [12].

According to the European Hypertension Guidelines, healthy blood pressure is defined as being below 120 mmHg for systolic and 80 mmHg for diastolic pressure [13, 14]. Hypertension, or high blood pressure, is characterized by a sustained elevation above these normal ranges. This condition is classified into different grades depending on the specific values, such as mild, moderate, or severe hypertension, all of which increase the risk of developing cardiovascular diseases like myocardial infarction, heart failure, stroke, and in severe cases, can lead to death [15–17].

Constant monitoring of blood pressure is essential to prevent and control hypertension, as its consequences can be devastating to health.

2.2 Ambulatory Blood Pressure Monitoring

Ambulatory Blood Pressure Monitoring (ABPM) is a non-invasive device designed to continuously measure and record a patient's blood pressure over a 24-h period. This tool provides a more detailed and accurate view of blood pressure and heart rate patterns throughout the day and night, offering valuable insights into cardiovascular health under normal daily activities and rest [18].

The device consists of a blood pressure cuff, which is placed on the patient's arm, and is connected to a portable monitoring unit. This monitor automatically records blood pressure readings at pre-programmed intervals, storing the data throughout the monitoring period. Once the study is completed, the collected information is transferred to a computer, where it is analyzed to identify abnormal patterns such as nocturnal hypertension or excessive variability that might not be apparent during routine office visits. This allows the physician to make a more accurate diagnosis and adjust treatment plans more effectively [19–23].

2.3 Nocturnal Blood Pressure Profile

The nocturnal blood pressure profile describes the pattern of blood pressure during nighttime rest, which can reveal issues with hypertension management or an increased cardiovascular risk. Normally, blood pressure decreases by 10–20% during sleep as part of a healthy circadian rhythm. If this drop is less than 10%, or if blood pressure rises during the night, it may indicate a disruption in blood pressure control and a heightened risk of cardiovascular events [24, 25].

There are several nocturnal blood pressure profiles:

- **Dipper profile**: Characterized by a 10–20% reduction in blood pressure, which is considered normal.

- **Non-dipper profile**: When the reduction is less than 10%, this profile may suggest poor hypertension control and potential cardiovascular risks.
- **Extreme dipper profile**: Where blood pressure drops by more than 20%.
- **Riser profile**: Where nighttime blood pressure exceeds daytime levels.

To assess the nocturnal blood pressure profile, calculate the ratio of daytime to nighttime readings using an Ambulatory Blood Pressure Monitoring (ABPM) device. This involves averaging systolic and diastolic readings for both day and night, and then dividing the average nighttime readings by the daytime averages. This ratio quantifies the variation in blood pressure between day and night, aiding in the identification of nocturnal hypertension patterns and enabling adjustments to treatment to improve blood pressure control and mitigate associated risks [26–28].

2.4 Interval Type-2 Fuzzy Logic

Interval Type-2 fuzzy logic can be considered an extension of Type-1 fuzzy logic, designed to manage a higher level of uncertainty. This is achieved by utilizing Type-2 membership functions, which incorporate the footprint of uncertainty (FOU). The FOU consists of two Type-1 membership functions, allowing membership in a fuzzy set to be expressed as an interval [29].

In a Type-2 membership function (IT2MF), the "lower scale" refers to reducing the maximum value of the lower membership function (LMF), while the "lower lag" involves shifting the LMF inward relative to the upper membership function (UMF). Both "lower scale" and "lower lag" values determine how the FOU of the function is shaped, with values ranging between 0 and 1. These adjustments allow for control over the shape and symmetry of the FOU [30, 31].

3 Methodology

This section details the analysis conducted on the fuzzy systems developed as part of a hybrid model [32, 33] combining neural networks and fuzzy logic, specialized in the early diagnosis of hypertension. This model integrates the learning capabilities of neural networks with the flexibility and uncertainty management of fuzzy systems, providing a robust tool for the precise identification of patterns associated with hypertension.

To assess the effectiveness of these fuzzy systems, experiments were conducted using various membership functions, which are crucial components in fuzzy logic for representing the degree of membership of a value to a fuzzy set. The structure of the first fuzzy system is designed with two primary inputs corresponding to the ratios of systolic blood pressure readings. Based on these inputs, the system outputs the nocturnal blood pressure profile. To represent these inputs, four membership

functions are employed with the linguistic values “Low,” “Normal,” “High,” and “Very High.” These terms describe different levels of blood pressure intensity.

The output of the fuzzy system is classified into four distinct membership functions, corresponding to the linguistic variables “Extreme Dipper,” “Dipper,” “Non-Dipper,” and “Riser.” These terms are used to define the different nocturnal blood pressure profiles, offering a clear interpretation of how blood pressure varies during the night. The design of the membership functions is illustrated in Fig. 1.

Regarding the configuration of the membership functions for both inputs and outputs, a value of 0.9 was applied to the lower scale. The “Lower Lag” parameter was manually adjusted to enhance the model’s accuracy.

The rules used in this fuzzy system are the following:

- If *Systolic is GreaterFall* and *Diastolic is GreaterFall,* then *Profile is ExtremeDipper*
- If *Systolic is Fall* and *Diastolic is Fall,* then *Profile is Dipper*
- If *Systolic is LowDecrease* and *Diastolic is LowDecrease,* then *Profile is NonDipper*
- If *Systolic is Increase* and *Diastolic is Increase*, then *Profile is Riser*

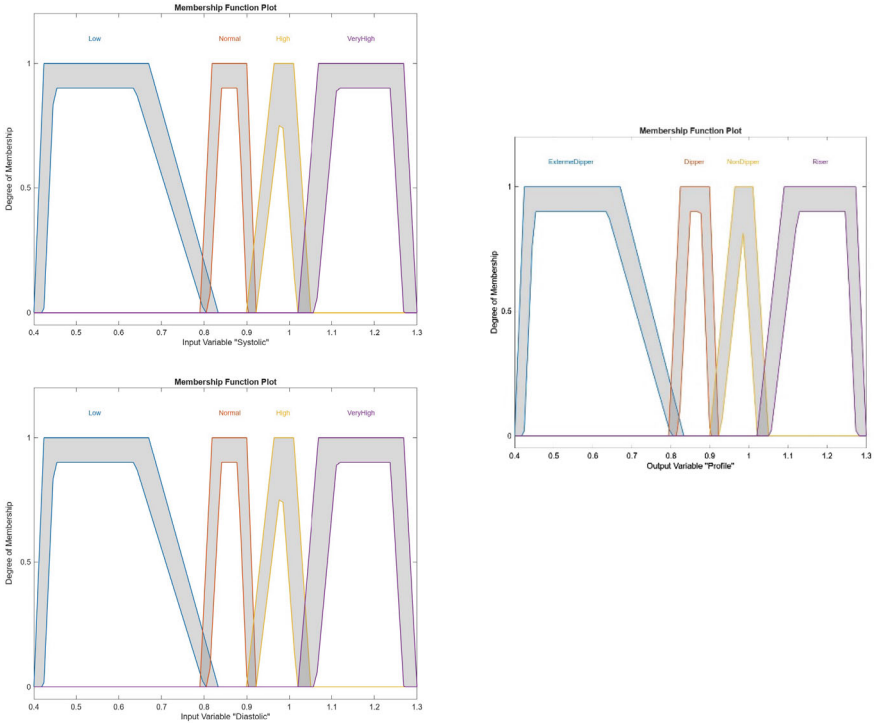


Fig. 1 Inputs and output for trapezoidal membership functions

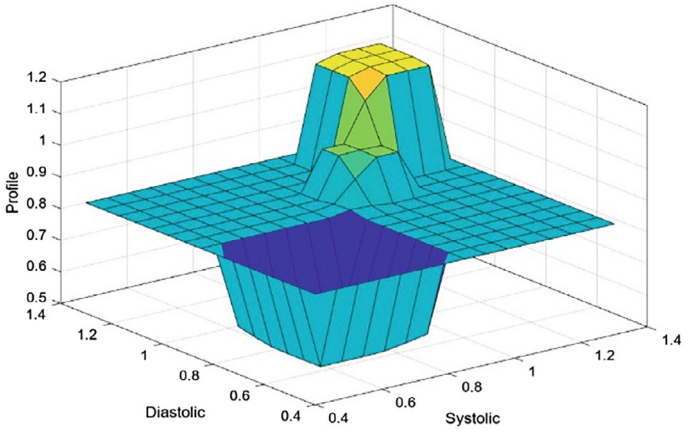


Fig. 2 Surface plot for trapezoidal fuzzy model

Figure 2 illustrates the surface of the fuzzy system, providing a detailed visualization of the rule-based system's behavior. This graphical representation facilitates the analysis of how different input combinations affect the system's output. Through this surface, one can observe how the fuzzy system processes and classifies the data to generate nocturnal blood pressure profiles, offering a valuable tool for understanding and fine-tuning the model according to the specific needs of the diagnosis.

In the design of the membership functions based on Gaussian functions, a configuration similar to that of the previously described fuzzy system is maintained, but with some key differences in parameterization. In this case, the lower scale value was adjusted to 0.85 for the Gaussian membership functions, unlike the value used in the fuzzy system with trapezoidal functions. Additionally, the lower lag value was set differently for each membership function, in contrast to the uniform approach applied in the previous design.

These adjustments to the lower scale and lower lag parameters significantly impact the behavior of the fuzzy system, as Gaussian functions provide a more flexible and refined representation of the data, better accommodating subtle variations in blood pressure. Figure 3 illustrates the detailed design of the fuzzy system using Gaussian membership functions, showing how these functions are distributed and adjusted to model nocturnal blood pressure profiles more accurately. This graphical representation offers a visual understanding of how changes in membership function parameters affect the system's performance and output.

Figure 4 shows the response surface of the fuzzy system with Gaussian membership functions. This visualization provides a detailed analysis of the rule system behavior and demonstrates how different configurations of the membership functions impact the system's output.

The response surface illustrates how combinations of inputs, specifically systolic and diastolic pressure ratios, affect the classification of the nocturnal blood pressure profile. Each point on the surface represents a potential output of the fuzzy system

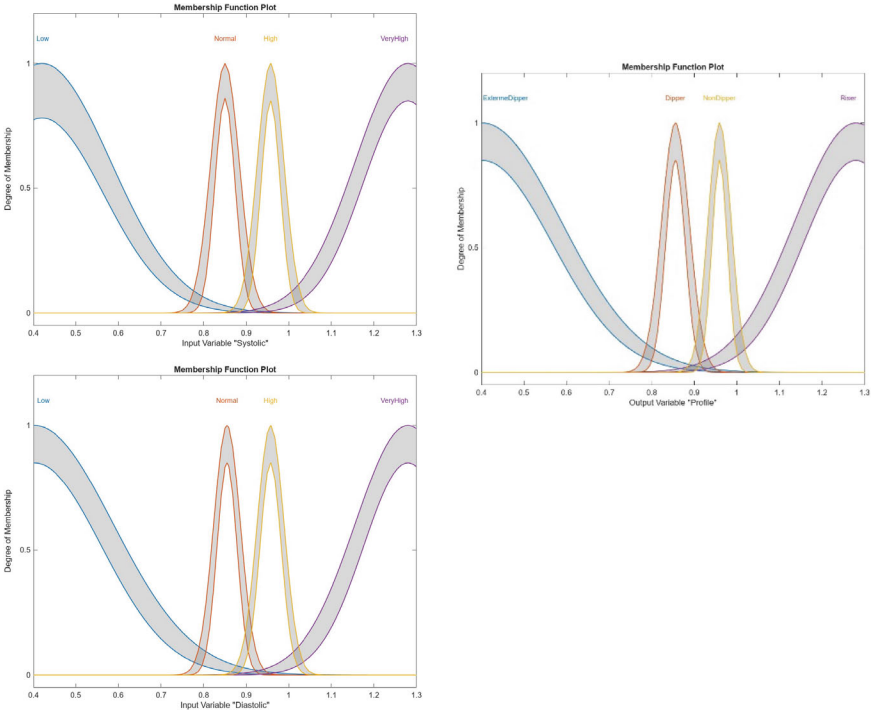


Fig. 3 Inputs and Output for Gaussian membership functions

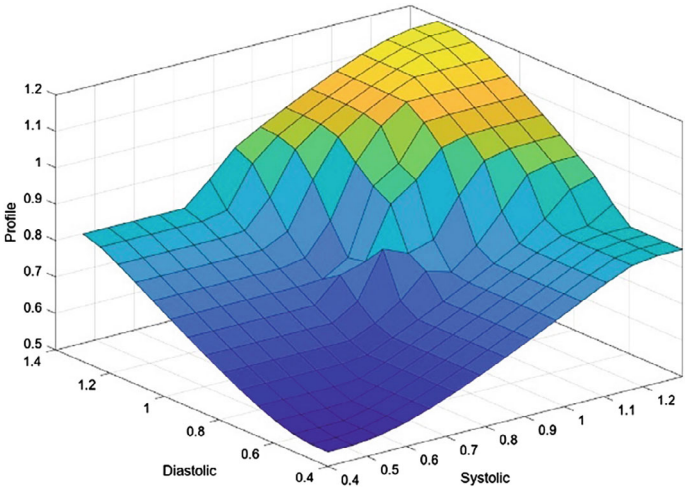


Fig. 4 Surface plot for Gaussian fuzzy model

for a specific set of inputs. This graphical representation is crucial for understanding how the system processes various linguistic variables and how adjustments in the Gaussian membership functions influence the system's response.

4 Results

In this section, we provide a detailed account of the tests conducted on the fuzzy inference systems to assess their performance in classifying nocturnal blood pressure profiles. The study involved a sample of 20 patients, each of whom wore an Ambulatory Blood Pressure Monitoring (ABPM) device. Continuous blood pressure readings were recorded throughout the day and night during the monitoring period, enabling a comprehensive collection of data.

Following the approach outlined in the literature review, daytime and nighttime readings were separated, and the ratio between systolic and diastolic pressures for each period was calculated. These ratios were then used as inputs for the fuzzy system. This methodology ensures that the system processes data according to the time of day, enhancing the accuracy of nocturnal profile identification.

Table 1 presents the results obtained using the IT2FS with Trapezoidal Membership Functions. The table is organized as follows: columns two through four show the averages of the daytime and nighttime systolic blood pressure readings, along with the calculated ratio. Columns five through seven present the averages of the daytime and nighttime diastolic blood pressure readings and their corresponding ratio. Finally, column eight presents the output generated by the fuzzy system, while column nine provides the translation of these results into the predefined nocturnal profile categories. It is worth mentioning that the result highlighted in bold is the incorrect classification.

This analysis enables the evaluation of the fuzzy system's ability to accurately classify nocturnal blood pressure profiles based on the obtained readings. It provides a clear understanding of how the system processes and translates data into meaningful categories for diagnostic purposes.

Table 2 shows the results obtained using the IT2FS with Gaussian Membership Functions. This table is organized similarly to Table 1, allowing for easy comparison and analysis of the data. It provides a structured overview of the average systolic and diastolic blood pressure readings taken during the day and night, as well as the calculated ratios for both periods. Additionally, it includes the results produced by the fuzzy system, offering a clear translation of these results into the nocturnal blood pressure profiles. This organized format not only facilitates the interpretation of the data but also helps in identifying patterns and trends that may be relevant for assessing blood pressure behavior in the analyzed patients. The results marked in bold indicate the erroneous classifications.

To assess the effectiveness of the fuzzy system, a confusion matrix was utilized. This matrix enables us to visualize and analyze the performance of the model in accurately classifying nocturnal blood pressure profiles. It records the predicted values

Table 1 Results obtained by the IT2 Trapezoidal fuzzy model

No	Systolic			Diastolic			Output	Profile
	Day	Night	Quot	Day	Night	Quot		
1	104	116	0.90	63	76	0.83	0.860	Dipper
2	103	126	0.82	61	80	0.76	0.621	Ex. dipper
3	107	118	0.91	64	68	0.94	0.976	Non-dipper
4	101	131	0.77	57	82	0.70	0.605	Ex. dipper
5	119	135	0.88	73	86	0.85	0.860	Dipper
6	115	123	0.93	56	67	0.84	0.850	Dipper
7	106	113	0.94	64	75	0.85	0.850	Dipper
8	118	126	0.94	71	85	0.84	0.850	Dipper
9	119	136	0.88	68	84	0.81	0.855	Dipper
10	93	96	0.97	61	61	1.00	0.978	Non-dipper
11	94	99	0.95	58	64	0.91	0.976	Non-dipper
12	96	111	0.86	55	69	0.80	0.855	Dipper
13	102	113	0.90	63	70	0.90	0.922	Non-dipper
14	110	121	0.91	61	71	0.86	0.855	Dipper
15	138	127	1.09	89	85	1.05	1.156	Riser
16	105	120	0.88	64	76	0.84	0.860	Dipper
17	110	118	0.93	49	56	0.88	0.850	Dipper
18	112	115	0.97	67	74	0.91	0.976	Non-dipper
19	100	108	0.93	56	62	0.90	0.976	Non-dipper
20	**107**	**104**	**1.03**	**62**	**63**	**0.98**	**0.976**	**Non-dipper**

against the actual values, making it easier to identify correct classifications and errors across the different categories.

Table 3 presents the confusion matrix for the results obtained using the IT2FS with Trapezoidal membership functions. It can be observed that, out of all the patients evaluated, only one incorrect result was obtained.

Table 4 presents the confusion matrix for the results obtained using IT2FS with Gaussian membership functions. Analyzing the results, it can be concluded that two patients were misclassified.

5 Conclusions and Future Work

In conclusion, the use of IT2FSs has demonstrated significant advantages in addressing uncertainty and imprecision, making them highly effective for medical classification tasks, particularly in nocturnal blood pressure profile assessment.

Table 2 Results obtained by the IT2 Gaussian fuzzy model

No	Systolic			Diastolic			Output	Profile
	Day	Night	Quot	Day	Night	Quot		
1	104	116	0.90	63	76	0.83	0.843	Dipper
2	103	126	0.82	61	80	0.76	0.653	Ex. dipper
3	107	118	0.91	64	68	0.94	0.929	Non-dipper
4	101	131	0.77	57	82	0.70	0.625	Ex. dipper
5	119	135	0.88	73	86	0.85	0.847	Dipper
6	**115**	**123**	**0.93**	**56**	**67**	**0.84**	**0.781**	**Ex. dipper**
7	106	113	0.94	64	75	0.85	0.807	Dipper
8	**118**	**126**	**0.94**	**71**	**85**	**0.84**	**0.778**	**Ex. dipper**
9	119	136	0.88	68	84	0.81	0.835	Dipper
10	93	96	0.97	61	61	1.00	0.984	Non-dipper
11	94	99	0.95	58	64	0.91	0.958	Non-dipper
12	96	111	0.86	55	69	0.80	0.800	Dipper
13	102	113	0.90	63	70	0.90	0.887	Dipper
14	110	121	0.91	61	71	0.86	0.843	Dipper
15	138	127	1.09	89	85	1.05	1.147	Riser
16	105	120	0.88	64	76	0.84	0.846	Dipper
17	110	118	0.93	49	56	0.88	0.875	Dipper
18	112	115	0.97	67	74	0.91	0.966	Non-dipper
19	100	108	0.93	56	62	0.90	0.931	Non-dipper
20	107	104	1.03	62	63	0.98	1.090	Riser

Table 3 Confusion matrix for results with Trapezoidal membership functions

	Ex. dipper	Dipper	Non-dipper	Riser
Real Ex. dipper	2	0	0	0
Real dipper	0	10	0	0
Real non dipper	0	0	6	1
Real riser	0	0	1	1

Table 4 Confusion matrix for results with Gaussian membership functions

	Ex. dipper	Dipper	Non-dipper	Riser
Real Ex. dipper	2	2	0	0
Real dipper	2	9	0	0
Real non dipper	0	0	5	0
Real riser	0	0	0	2

Quantitatively, our study showed that the IT2FS with trapezoidal membership functions achieved a classification accuracy of 96%, with only one misclassification. In comparison, the IT2FS using Gaussian membership functions yielded slightly lower performance, with two misclassifications and an accuracy of 92%. These results indicate that while both membership functions performed similarly, the trapezoidal function offers better precision for this specific medical application.

The high accuracy of the trapezoidal IT2FS suggests that it can be a valuable tool for physicians in improving hypertension diagnosis and managing nocturnal blood pressure variations. Furthermore, these findings underscore the potential of IT2FS to enhance the reliability of fuzzy-based classification models in real-world medical applications.

Future work will focus on further optimizing the system through bio-inspired algorithms to improve precision and adaptability. Additionally, we aim to explore generalized type-2 fuzzy systems and expand the study to a larger cohort of patients, which will strengthen the robustness of the model and its applicability to diverse populations.

References

1. S. Thudumu, P. Branch, J. Jin, and J. (Jack) Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *J Big Data*, vol. 7, no. 1, pp. 1–30, Dec. 2020
2. A. Jabbar, P. Akhtar, and S. Dani, "Real-time big data processing for instantaneous marketing decisions: A problematization approach," *Industrial Marketing Management*, vol. 90, pp. 558–569, Oct. 2020
3. H. Mo, Y. Meng, F. Y. Wang, and D. Wu, "Interval Type-2 Fuzzy Hierarchical Adaptive Cruise Following-Control for Intelligent Vehicles," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 9, pp. 1658–1672, Sep. 2022
4. A. Beke and T. Kumbasar, "More Than Accuracy: A Composite Learning Framework for Interval Type-2 Fuzzy Logic Systems," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 3, pp. 734–744, Mar. 2023
5. D. J. Singh, N. K. Verma, A. K. Ghosh, and A. Malagaudanavar, "An application of interval type-2 fuzzy model based control system for generic aircraft," *Appl Soft Comput*, vol. 121, p. 108721, 2022
6. A. Safari, R. Hosseini, and M. Mazinani, "A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction," *J Biomed Inform*, vol. 123, pp. 1–10, Nov. 2021
7. A. Safari, R. Hosseini, and M. Mazinani, "A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction," *J Biomed Inform*, vol. 123, pp. 1–10, Nov. 2021
8. B. Menİz and E. M. Özkan, "Vaccine selection for COVID-19 by AHP and novel VIKOR hybrid approach with interval type-2 fuzzy sets," *Eng Appl Artif Intell*, vol. 119, pp. 1–13, Mar. 2023
9. H. Azadi, M.-R. Akbarzadeh-T, H.-R. Kobravi, and A. Shoeibi, "Robust Voice Feature Selection Using Interval Type-2 Fuzzy AHP for Automated Diagnosis of Parkinson's Disease," *IEEE/ACM Trans Audio Speech Lang Process*, vol. 29, pp. 2792–2802, 2021

10. R. Sharma and A. Kumar, "Optimal Interval Type-2 Fuzzy Logic Control Based Closed-Loop Regulation of Mean Arterial Blood Pressure Using the Controlled Drug Administration," *IEEE Sens J*, vol. 22, no. 7, pp. 7195–7207, 2022
11. A. E. Schutte, A. Kollias, and G. S. Stergiou, "Blood pressure and its variability: classic and novel measurement techniques," Oct. 01, 2022, *Nature Research*
12. J. Redon, G. Pichler, and F. Martinez, "Blood Pressure Control in Europe and Elsewhere," in *Manual of Hypertension of the European Society of Hypertension*, CRC Press, 2019, pp. 25–30
13. G. Mancia, G. Grassi, K. P. Tsioufis, A. F. Dominiczak, and E. Agabiti Rosei, *Manual of Hypertension of the European Society of Hypertension.* CRC Press, 2019.
14. G. S. Stergiou *et al.*, "2021 European Society of Hypertension practice guidelines for office and out-of-office blood pressure measurement," *J Hypertens*, vol. 39, no. 7, pp. 1293–1302, Jul. 2021
15. G. C. Oh and H. J. Cho, "Blood pressure and heart failure," Jan. 02, 2020, *BioMed Central Ltd*.
16. F. D. Fuchs and P. K. Whelton, "High Blood Pressure and Cardiovascular Disease," Feb. 01, 2020, *Lippincott Williams and Wilkins.*
17. A. N. Desai, "High Blood Pressure," *JAMA*, vol. 324, no. 12, pp. 1254–1255, Sep. 2020
18. Q. F. Huang *et al.*, "Ambulatory Blood Pressure Monitoring to Diagnose and Manage Hypertension," Feb. 01, 2021, *Lippincott Williams and Wilkins.*
19. K. Kario, "Home blood pressure monitoring: Current status and new developments," *Am J Hypertens*, vol. 34, no. 8, pp. 783–794, Aug. 2021
20. C. Pena-Hernandez, K. Nugent, and M. Tuncel, "Twenty-Four-Hour Ambulatory Blood Pressure Monitoring," *J Prim Care Community Health*, vol. 11, pp. 1–8, 2020
21. R. A. Sánchez *et al.*, "Ambulatory blood pressure monitoring over 24 h: A Latin American Society of Hypertension position paper—accessibility, clinical use and cost effectiveness of ABPM in Latin America in year 2020," *J Clin Hypertens*, vol. 22, no. 4, pp. 527–543, Apr. 2020
22. B. C. Jaeger *et al.*, "Number and timing of ambulatory blood pressure monitoring measurements," *Hypertension Research*, vol. 44, no. 12, pp. 1578–1588, 2021
23. E. Vinyoles *et al.*, "Role of ambulatory blood pressure on prediction of cardiovascular disease. A cohort study," *J Hum Hypertens*, vol. 37, no. 4, pp. 279–285, 2023
24. P. Anyfanti, A. Malliora, A. Chionidou, K. Mastrogiannis, A. Lazaridis, and E. Gkaliagkousi, "Clinical Significance of Nocturnal Hypertension and Nighttime Blood Pressure Dipping in Hypertension," *Curr Hypertens Rep*, vol. 26, no. 2, pp. 69–80, 2024
25. F. P. Cappuccio, "The role of nocturnal blood pressure and sleep quality in hypertension management," *European Cardiology Review*, vol. 15, p. 1, 2020
26. K. Kario and B. Williams, "Nocturnal Hypertension and Heart Failure: Mechanisms, Evidence, and New Treatments," *Hypertension*, vol. 78, no. 3, pp. 564–577, Sep. 2021
27. B. C. Jaeger *et al.*, "Development of Predictive Equations for Nocturnal Hypertension and Nondipping Systolic Blood Pressure," *J Am Heart Assoc*, vol. 9, no. 2, pp. 1–48, Jan. 2020
28. K. Kario *et al.*, "Nighttime Blood Pressure Phenotype and Cardiovascular Prognosis: Practitioner-Based Nationwide JAMP Study," *Circulation*, vol. 142, no. 19, pp. 1810–1820, Nov. 2020
29. E. Ontiveros, P. Melin, and O. Castillo, "Comparative study of interval Type-2 and general Type-2 fuzzy systems in medical diagnosis," *Inf Sci (N Y)*, vol. 525, pp. 37–53, 2020
30. M. Dirik, O. Castillo, and A. F. Kocamaz, "Visual-servoing based global path planning using interval type-2 fuzzy logic control," *Axioms*, vol. 8, no. 2, Jun. 2019
31. D. Wu and J. M. Mendel, "Recommendations on designing practical interval type-2 fuzzy systems," *Eng Appl Artif Intell*, vol. 85, pp. 182–193, 2019
32. P. Melin, I. Miramontes, and G. Prado-Arechiga, "A hybrid model based on modular neural networks and fuzzy systems for classification of blood pressure and hypertension risk diagnosis," *Expert Syst Appl*, vol. 107, pp. 146–164, 2018
33. I. Miramontes, P. Melin, and G. Prado-Arechiga, "Fuzzy System for Classification of Nocturnal Blood Pressure Profile and Its Optimization with the Crow Search Algorithm," in *Soft Computing Applications*, V. E. Balas, L. C. Jain, M. M. Balas, and S. N. Shahbazova, Eds., Cham: Springer International Publishing, 2021, pp. 23–34.

Glioma Segmentation in T1-Weighted MRI Images Using Deep Learning

Fernando Morquecho Lara, Oscar Montiel, Miguel López Montiel, and Moisés Sánchez Adame

Abstract Precise segmentation of gliomas in magnetic resonance imaging (MRI) is essential for enhancing the diagnosis and treatment of brain cancer. This study employs advanced deep learning techniques, specifically a modified U-Net convolutional neural network, to identify and segment glioma-affected regions in T1-weighted MRI images. The model demonstrated high accuracy in segmentation tasks, outperforming existing methods in accuracy and processing time. Trained and evaluated on a comprehensive clinical dataset, our approach shows substantial promise in assisting specialists in analyzing brain tumors more effectively and optimizing radiology workflows. These findings highlight the innovative potential of deep learning in medical imaging.

Introduction

Gliomas are among the most prevalent and aggressive brain tumors, accounting for a significant portion of malignant brain cancers [1]. These tumors pose substantial challenges in clinical practice due to their rapid progression, high recurrence rates, and their ability to infiltrate surrounding brain tissue. Accurate detection and segmentation of gliomas are critical for effective treatment planning, monitoring disease progression, and improving patient outcomes [2]. Furthermore, precise

F. Morquecho Lara · O. Montiel (✉) · M. S. Adame
Instituto Politécnico Nacional-CITEDI, Tijuana, B.C., Mexico
e-mail: oross@ipn.mx

F. Morquecho Lara
e-mail: fmorquecho@citedi.mx

M. S. Adame
e-mail: msanchez@citedi.mx

M. L. Montiel
ITJ Tijuana, Blvd. Salinas 10485-Interior 1403, Tijuana, B.C., Mexico
e-mail: miguel.lopez@itj.com

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_4

tumor delineation is essential for guiding surgical resections, radiotherapy planning, and evaluating therapeutic responses, making segmentation a cornerstone in neuro-oncology.

Despite the pivotal role of glioma segmentation in clinical workflows, the manual process remains fraught with limitations. Manual segmentation of MRI scans, particularly T1-weighted sequences, is inherently challenging due to the complexity and variability of glioma appearances. This task requires significant expertise, is time-intensive, and is prone to inconsistencies stemming from inter- and intra-observer variability [3]. Even experienced radiologists may encounter difficulties in distinguishing tumor boundaries from surrounding edema or necrotic regions, which often exhibit overlapping intensity profiles [4]. These challenges can lead to delays in diagnosis and treatment planning, ultimately impacting patient care outcomes.

Moreover, manual segmentation does not scale well with the increasing availability of imaging data, creating bottlenecks in clinical and research settings. As advancements in imaging technologies enable the acquisition of high-resolution scans, the reliance on manual segmentation further exacerbates the workload for clinicians and radiologists [5]. This underscores the urgent need for automated solutions that not only enhance segmentation accuracy but also reduce the time and effort required for data processing.

Automated segmentation methods leveraging deep learning have emerged as transformative solutions to address these limitations. Convolutional Neural Networks (CNNs) have shown remarkable success in medical image analysis by learning hierarchical features directly from data, thereby eliminating the dependency on handcrafted feature extraction [5]. These methods deliver consistent and reproducible results, mitigating the challenges of inter-observer variability and improving the standardization of tumor delineation [2]. Additionally, the ability of CNN-based models to process large datasets efficiently aligns with the growing demand for scalable and robust diagnostic tools in clinical practice.

In this work, we focus on the U-Net architecture, a widely recognized CNN model specifically designed for biomedical image segmentation [5]. The encoder-decoder structure of U-Net, combined with skip connections, facilitates the preservation of spatial information while capturing both global and local contexts. This design is particularly advantageous for segmenting gliomas, where precise boundary delineation is critical for distinguishing tumor regions from healthy tissue. The U-Net model's success in handling complex structures, even with limited annotated data, makes it an ideal candidate for addressing the unique challenges of glioma segmentation [6].

By automating the segmentation process, this work aims to provide clinicians with a faster, more reliable, and scalable solution that overcomes the inherent limitations of manual methods. Through the use of publicly available datasets, such as those provided by the Brain Tumor Segmentation (BraTS) challenge [4], and the application of robust performance metrics like precision, recall, Dice Coefficient, and F1 score, this study demonstrates the potential of U-Net-based architectures to revolutionize glioma diagnosis and treatment workflows (Fig. 1).

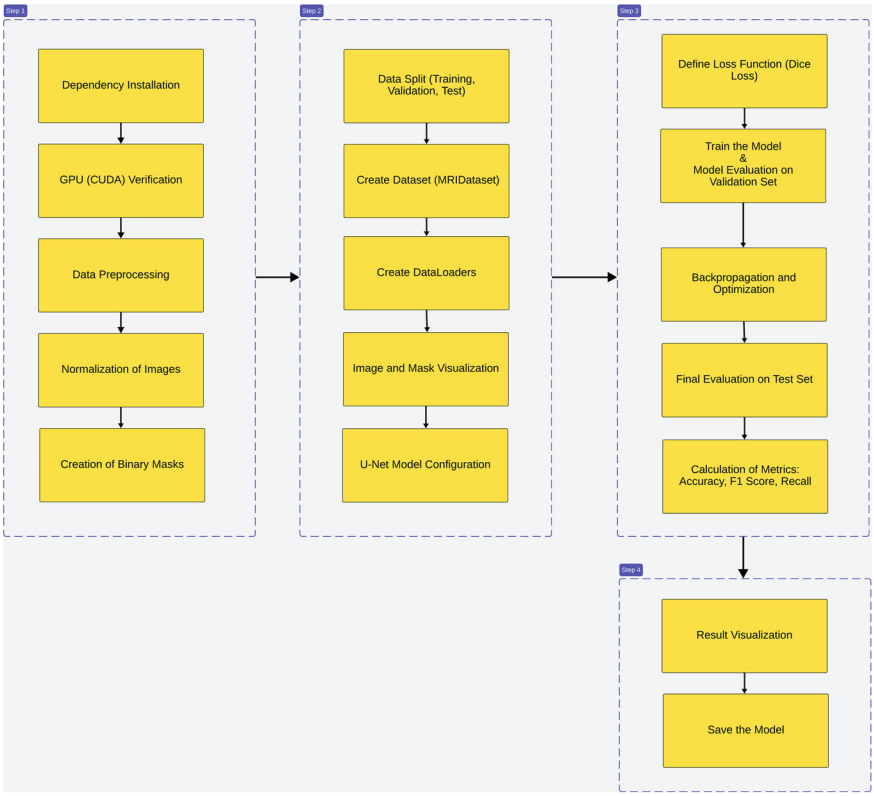


Fig. 1 Workflow for glioma segmentation using T1-weighted MRI images. The diagram outlines the key steps, including dependency installation, data preprocessing, model configuration, training, evaluation, and final result visualization

Related Work

The segmentation of brain tumors in magnetic resonance imaging (MRI) has significantly advanced with the adoption of deep learning techniques. Prior to the widespread use of these approaches, traditional machine learning methods were commonly applied. However, these methods often exhibited limitations in precision, computational efficiency, and their reliance on handcrafted features, which restricted their ability to capture complex patterns in medical images.

The introduction of convolutional neural networks (CNNs) revolutionized the field by enabling automatic medical image segmentation with remarkable accuracy. One of the most influential architectures in this domain is the U-Net proposed by Ronneberger et al. [5]. U-Net's success lies in its symmetric encoder-decoder structure, which combines a contracting path to extract contextual information and an expanding path to achieve precise localization. Including skip connections allows for

the retention of fine-grained details lost during downsampling. This design enables U-Net to produce high-quality segmentation maps, even when trained on limited annotated data, making it particularly valuable in biomedical applications where labeled datasets are often scarce [5].

Since its introduction, U-Net variants have been developed to address the challenges in brain tumor segmentation. For instance, Al Nasim et al. [7] proposed an enhanced U-Net model trained on the BraTS (Brain Tumor Segmentation Challenge) dataset. Their model demonstrated the capability to segment complex tumor regions, including necrotic tissue, edema, and enhancing tumor areas, achieving Dice scores of 0.8717 for necrotic tissue, 0.9506 for edema, and 0.9427 for enhancing regions. These results emphasize the robustness of the U-Net architecture for brain tumor segmentation tasks [7].

Researchers have explored multimodal approaches that integrate different MRI sequences to further enhance segmentation precision. For example, combining T1, T1c, T2, and FLAIR MRI modalities with CNNs enables the models to leverage complementary information across modalities. This multimodal integration improves the differentiation between tumor tissues and healthy regions, which is essential for accurate tumor delineation and treatment planning [4].

In recent years, the focus has shifted toward modern architectures like attention mechanisms and transformers, which provide enhanced feature representation and robustness in brain tumor segmentation. These models address the high variability in tumor shapes, sizes, and imaging quality, which remains a challenge for traditional CNN-based approaches Nguyen-Tat et al. [8]. introduced a hybrid approach combining U-Net with attention mechanisms and transformer-based modules, resulting in improved segmentation accuracy and resilience to variability in MRI datasets [8]. The addition of attention mechanisms allows the network to focus on the most relevant regions in the image, while transformer-based modules enhance global context awareness, a critical aspect for segmenting irregularly shaped tumors.

While U-Net and its variants excel in capturing local and fine-grained details, transformer-based models and hybrids provide a complementary advantage by capturing long-range dependencies and contextual relationships across the image. For instance, models like TransUNet [9] and the hybrid U-Net-Transformer proposed by Nguyen-Tat et al. [8]. Demonstrate superior performance in datasets with high variability. These models mitigate overfitting and achieve better generalization due to their ability to aggregate information from broader spatial regions.

Integrating transformers into segmentation tasks marks a significant leap forward but also presents challenges such as increased computational complexity and the need for larger annotated datasets. Future research should optimize these architectures for resource-constrained environments and explore semi-supervised or unsupervised learning techniques to address data scarcity.

Data Distribution

For this work, we used a publicly available dataset from Kaggle, which contains T1-weighted MRI images labeled for glioma segmentation. The dataset was divided into three subsets to ensure an effective training, validation, and evaluation process: 60% for training, 20% for validation, and 20% for testing [1]. This division allowed the model to learn efficiently, adjust its parameters, and generalize effectively to unseen data.

Maintaining a balanced dataset across classes was essential to mitigate bias during training, ensuring equal representation for both categories. Figure 2 summarizes the number of images in each class, where the balanced dataset distribution can be visualized.

- **No Tumor**: 3,925 images
- **Tumor**: 3,925 images.

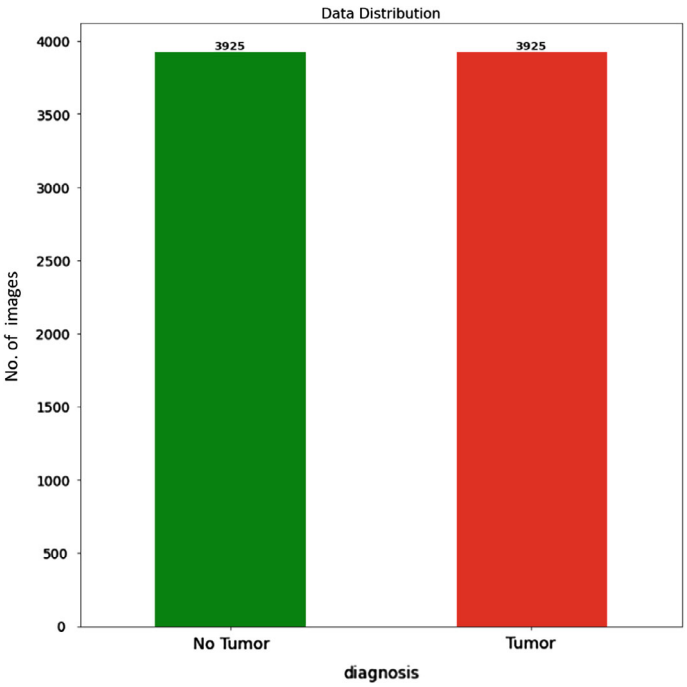


Fig. 2 Balanced data distribution showing an equal number of images for 'No Tumor' and 'Tumor' categories. Each class contains 3,925 images, ensuring a fair and unbiased training process. Balanced data distribution showing an equal number of images for 'No Tumor' and 'Tumor' categories. Each class contains 3,925 images, ensuring a fair and unbiased training process

A balanced dataset prevents bias in the learning process, ensuring that the model can generalize well across both categories. Figure 2 highlights this equal representation, which is critical for achieving robust segmentation performance.

Preprocessing Steps

Data preprocessing is a critical step to standardize input images, ensuring consistent quality and robustness during model training. The following preprocessing techniques were applied, significantly enhancing the model's ability to segment gliomas accurately:

- **Resizing**: All images were resized to a uniform resolution of 256×256. This resolution strikes a balance between preserving anatomical details and reducing computational costs, making the images compatible with the U-Net architecture [5]. Resizing ensures that input dimensions are standardized, preventing issues related to varying image sizes during training.
- **Normalization**: The pixel intensity values of all images were normalized to the range **[0, 1]** using the formula:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}},$$

  where X is the pixel intensity. Normalization reduces the variance in image brightness, ensuring numerical stability during training and accelerating model convergence. This step is particularly important for enhancing the model's ability to generalize across datasets with varying intensity ranges [6].
- **Contrast Enhancement**: To improve tumor visibility, **histogram equalization** was applied. This technique enhances image contrast, allowing the model to identify tumor boundaries more effectively by highlighting subtle variations in intensity. The improved contrast ensures that glioma regions are more distinguishable, which directly improves the segmentation accuracy [2] (Fig. 3).

Model Architecture: U-Net

For the glioma segmentation task, the U-Net architecture was selected due to its established effectiveness in biomedical image segmentation [5]. The U-Net model is particularly well-suited for medical imaging due to its ability to preserve spatial details while capturing contextual information. It combines an encoder-decoder design with skip connections, which are critical for reconstructing fine-grained segmentation masks. The key components of the U-Net architecture are:

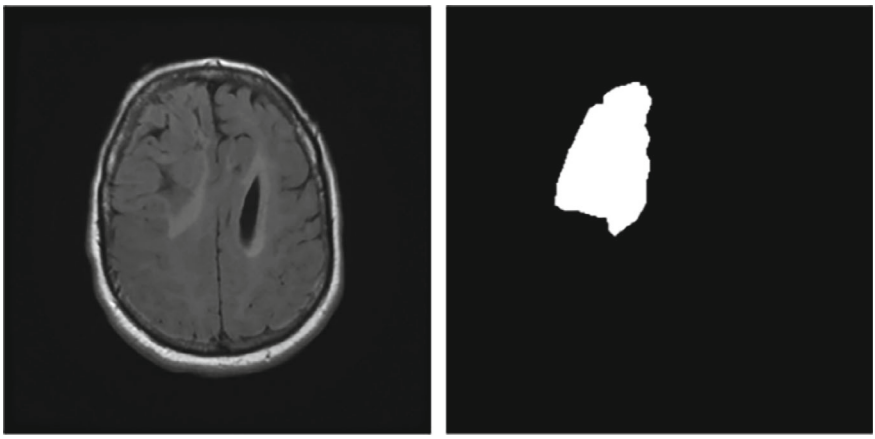


Fig. 3 Comparison between a raw MRI image (left) and its preprocessed version (right). The preprocessed image shows improved contrast and normalized intensity values, enhancing tumor visibility and aiding the segmentation process

- **Encoder**: The encoder extracts hierarchical features from the input images through a series of convolutional layers and max-pooling operations. These layers progressively reduce the spatial dimensions while increasing the depth of feature maps.
- **Decoder**: The decoder reconstructs the segmentation masks by progressively upsampling the feature maps. This restores the spatial resolution while integrating learned features from the encoder.
- **Skip Connections**: Skip connections link corresponding layers in the encoder and decoder, enabling the network to retain spatial details that are often lost during downsampling. This mechanism is essential for achieving precise boundary delineation in segmentation tasks [5].

Figure 4 illustrates the U-Net architecture, highlighting its encoder-decoder structure and the role of skip connections in preserving spatial details.

Data Augmentation

To address the limitations of small datasets and improve the model's generalization ability, data augmentation techniques were applied. These transformations introduced variability in the training data by simulating realistic anatomical changes and imaging variations. The applied augmentation techniques include:

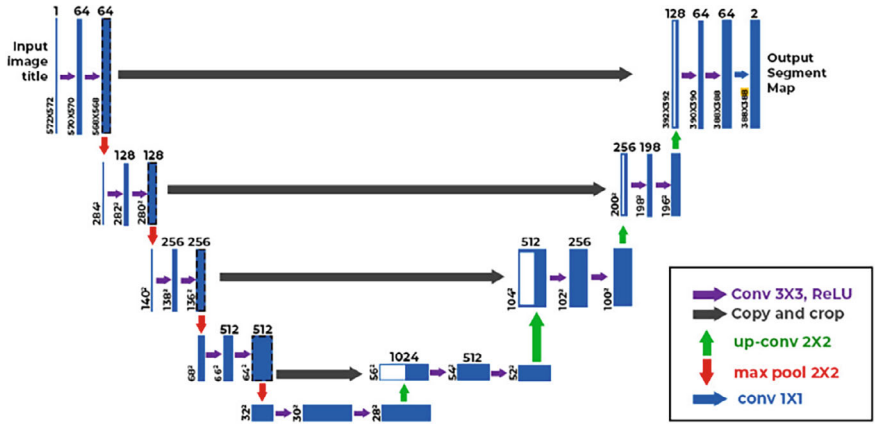


Fig. 4 The U-Net architecture showing the encoder-decoder design with skip connections, which are critical for achieving accurate segmentation results

- **Random Flipping**: Horizontal and vertical flips were applied to simulate different orientations of the brain.
- **Random Rotations**: Rotations within a range of $-20°$ to $+20°$ were introduced to account for slight misalignments in MRI scans.
- **Elastic Deformations**: Realistic deformations were simulated to account for anatomical variability in the brain [1, 5].

Preprocessing Steps

The following preprocessing steps were applied sequentially to standardize the input data:

- **Normalization**: Pixel intensities were normalized to the range [0, 1] to stabilize the learning process and accelerate model convergence [6].
- **Histogram Equalization**: Applied to improve the visibility of glioma regions by enhancing image contrast [2].
- **Z-score Normalization**: Adjusted pixel intensities to have a mean of 0 and a standard deviation of 1, further improving numerical stability during training [4].
- **Patch Extraction**: Large 3D MRI scans were divided into smaller patches to ensure that fine tumor regions were not overlooked during training [10].

Model Selection and Training

The U-Net model was trained using the Dice Loss function, which measures the degree of overlap between predicted segmentation masks and ground truth. The Dice Loss is particularly effective for medical image segmentation, where accurate boundary delineation is critical and class imbalance is common. The loss is defined as:

$$\text{Dice Loss} = 1 - \frac{2 \cdot |P \cap G|}{|P| + |G|},$$

where P represents the predicted segmentation mask and G is the ground truth [6].

Performance Metrics

The performance of the model was evaluated using standard segmentation metrics to provide a comprehensive understanding of its accuracy and reliability [4, 10]:

- **Accuracy**: Measures the proportion of correctly classified pixels. It is a standard metric for evaluating segmentation tasks in medical imaging [5].
- **Precision**: Assesses the model's ability to avoid false positives by measuring the fraction of correctly segmented tumor regions among predicted positives. High precision is critical in reducing misclassifications [1].
- **Recall**: Evaluates the model's sensitivity to correctly identify tumor regions. Recall ensures that fewer glioma regions are missed, which is particularly important in medical diagnoses [2].
- **F1 Score**: Combines precision and recall into a single metric, offering a balanced evaluation of segmentation performance. It is widely used for handling class imbalances in medical imaging [4].

To facilitate an intuitive understanding of the segmentation results, the output masks were visualized and overlaid on the original MRI scans. This visualization enabled a qualitative assessment of the model's ability to identify glioma-affected regions accurately.

Training Process and Dice Loss Function

The training process for the enhanced U-Net model was meticulously designed to ensure robust performance and accuracy in glioma segmentation. Central to this methodology was the Dice loss function, a metric tailored for medical imaging tasks. This function directly measures the overlap between the predicted segmentation

masks and the ground truth, enabling precise boundary delineation of glioma regions-an essential requirement in medical applications [6].

Dice Loss Function

Mathematically, the Dice loss is derived from the Dice coefficient, which evaluates the similarity between two sets, P (predicted mask) and G (ground truth mask). The Dice coefficient is defined as:

$$\text{Dice Coefficient} = \frac{2 \cdot \sum_{i=1}^{N} P_i G_i}{\sum_{i=1}^{N} P_i^2 + \sum_{i=1}^{N} G_i^2}$$

where:

- P_i is the predicted value for pixel i,
- G_i is the ground truth value for pixel i,
- N is the total number of pixels.

The Dice loss, used as the optimization objective, is expressed as:

$$\text{Dice Loss} = 1 - \text{Dice Coefficient}$$

This loss function is particularly effective for segmentation tasks due to its focus on overlapping regions, which helps mitigate issues arising from class imbalances, such as small tumor regions relative to healthy tissues (Fig. 5).

Training Configuration

To optimize the model's parameters, the Adam optimizer was employed, known for its efficiency in handling complex datasets and its adaptive learning rate adjustment. The optimizer was configured with a learning rate of 0.001, a value chosen after

```
def dice_loss(pred, target, smooth=1.0):
    pred = pred.view(-1)
    target = target.view(-1)
    intersection = (pred * target).sum()
    return 1 - (2. * intersection + smooth) / (pred.sum() + target.sum() + smooth)


optimizer = torch.optim.Adam(model.parameters(), lr=1e-4)
```

Fig. 5 Code implementation of the Dice Loss function used during training to optimize segmentation performance

preliminary experimentation. This value ensured stable convergence during training without causing oscillations or overshooting the optimal solution [6]. A batch size of 32 was selected to strike a balance between computational efficiency and model performance, allowing the model to process sufficient samples per iteration while staying within memory constraints.

Training was conducted over 10 epochs, a duration sufficient for the model to learn the critical features within the dataset while minimizing overfitting. To further mitigate overfitting and improve generalization, an early stopping mechanism was implemented. Early stopping monitored the validation loss during training and halted the process automatically if no significant improvement was observed for several consecutive epochs. This mechanism not only reduced computational costs but also ensured that the model generalized well to unseen data by avoiding overfitting [10].

The combined use of Adam optimizer, a carefully tuned learning rate, and early stopping enabled the model to achieve consistent and robust performance, laying a solid foundation for effective segmentation of glioma regions.

Evaluation Metrics and Results

The model's performance was evaluated using standard segmentation metrics, including the Dice Coefficient, F1 Score, Precision, and Recall. These metrics provided a balanced assessment of the model's ability to delineate glioma regions accurately while generalizing effectively to unseen data [4, 6].

The choice of Dice Loss as the optimization function proved particularly effective in addressing the challenges of medical image segmentation, such as class imbalance and precise boundary delineation. Metrics derived from Dice Loss offered comprehensive insights into segmentation performance:

- **Dice Coefficient**: Measures the degree of overlap between predicted segmentation masks and ground truth.
- **F1 Score**: Combines Precision and Recall, offering a balanced evaluation of the model's segmentation quality.
- **Precision**: Assesses the model's ability to avoid false positives.
- **Recall**: Measures the model's sensitivity to identifying all true positives.

The structured training procedure enabled the model to handle the inherent variability and complexity of medical imaging data effectively. By leveraging the Dice Loss and early stopping mechanisms, the U-Net model demonstrated robust performance, achieving high accuracy and generalization across validation datasets. This foundation underscores its potential for clinical applications in the diagnosis and treatment planning of gliomas.

Mathematical Formulation of Metrics

To evaluate the performance of the segmentation model, we used standard metrics such as F1 score, precision, recall, and Dice coefficient. These metrics provide a comprehensive understanding of the model's effectiveness in identifying glioma regions accurately [4, 6, 10].

F1 Score

The F1 score combines precision and recall into a single metric, serving as their harmonic mean. It is defined as:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision

Precision measures the proportion of correctly identified tumor regions among all predicted tumor regions. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where:

- TP: True Positives, i.e., correctly identified tumor regions.
- FP: False Positives, i.e., healthy regions incorrectly classified as tumors.

Recall (Sensitivity)

Recall quantifies the ability of the model to identify all actual tumor regions. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where:

- FN: False Negatives, i.e., tumor regions incorrectly classified as healthy.

Dice Coefficient

The Dice coefficient measures the overlap between the predicted segmentation mask and the ground truth. It is defined as:

$$\text{Dice Coefficient} = \frac{2 \cdot |\text{Prediction} \cap \text{Ground Truth}|}{|\text{Prediction}| + |\text{Ground Truth}|}$$

Or equivalently, in terms of TP, FP, and FN:

$$\text{Dice Coefficient} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}$$

Summary

These metrics provide a balanced evaluation of the model's segmentation capabilities:

- Precision focuses on minimizing false positives.
- Recall emphasizes the minimization of false negatives.
- F1 Score balances both precision and recall, offering a single comprehensive metric.
- Dice Coefficient measures the similarity between the predicted and actual segmentation regions, ensuring the model captures both tumor boundaries and volumes effectively.

Validation Metrics: The model's performance during validation is summarized in Table 1.

Table 1 Performance metrics during validation

Metric	Training value	Validation value
Precision	0.880	0.865
Recall	0.850	0.840
F1 score	0.865	0.852
Dice coefficient	0.870	0.860

Discussion of Results

The results indicate that the segmentation model performs well in distinguishing glioma regions from healthy brain tissue. The precision of 0.865 demonstrates that the model minimizes false positives effectively, which is critical in medical imaging to reduce unnecessary diagnoses. The recall of 0.840 shows that the model is capable of identifying the majority of tumor regions, although some subtle regions might have been missed.

The F1 score of 0.852 balances the precision and recall, highlighting the model's robustness in handling class imbalances in the dataset. Additionally, the Dice coefficient of 0.860 reflects a high level of overlap between the predicted and ground truth segmentation masks, validating the accuracy of the model's predictions.

These results were achieved due to several factors:

- **Preprocessing Techniques**: Effective normalization and data augmentation improved the generalization ability of the model.
- **Dice Loss Function**: The optimization using Dice Loss prioritized overlap regions, enhancing segmentation accuracy.
- **Model Architecture**: The U-Net model's encoder-decoder structure facilitated precise feature extraction and reconstruction of tumor regions.
- **Validation Strategy**: Early stopping prevented overfitting, ensuring that the model generalized well to unseen data.

Future work could explore advanced techniques like ensemble modeling and semi-supervised learning to address the subtle cases where recall was slightly lower. These improvements could further enhance the reliability of the segmentation tool in clinical applications (Fig. 6).

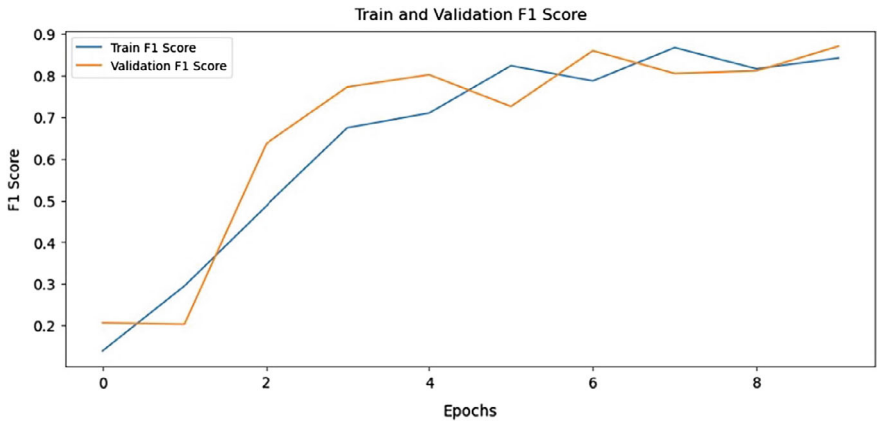


Fig. 6 Visualization of the F1 Score over epochs, demonstrating consistent improvements during training and validation. The graph includes clearly labeled axes, a legend, and grid lines for better readability

Training and Validation Results Summary

The training and validation process demonstrated the model's capacity to learn and generalize from the dataset effectively. The sharp increase in accuracy during the initial epochs reflects the model's ability to quickly identify essential patterns, aided by preprocessing techniques like data augmentation and normalization. By epoch 4, the training and validation accuracy stabilized, as shown in Fig. 7. This stabilization, along with consistent F1 scores and Dice coefficients, indicates that the model achieved robust performance while mitigating significant overfitting, thanks to the enhanced U-Net architecture and deep supervision strategies.

Comparison with Existing Models

To further contextualize the results, the performance of the proposed U-Net model was compared with state-of-the-art architectures from recent literature:

- The **VGG-UNet** achieved a higher F1 score of 0.879, leveraging ensemble techniques to mitigate biases in segmentation predictions [11].
- The **nn-UNet** demonstrated similar performance with a Dice coefficient of 0.870, aided by its automated configuration and deep supervision strategies [12].

While the proposed U-Net achieved competitive results, future iterations could incorporate ensemble learning and attention mechanisms to enhance performance further.

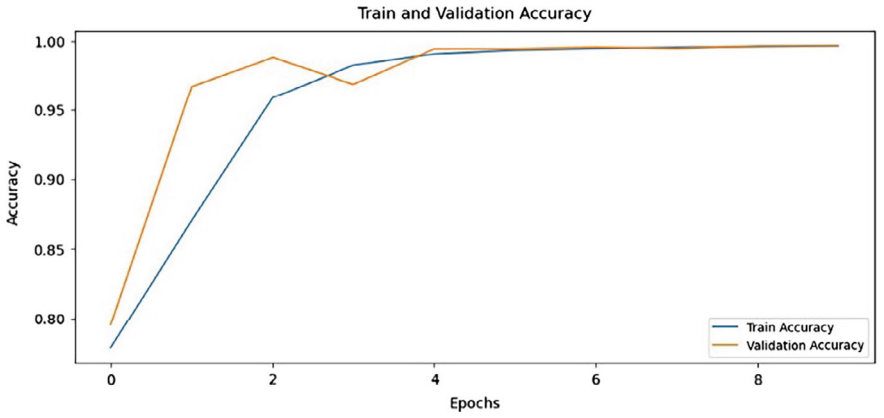


Fig. 7 Graph illustrating training and validation loss over epochs. The stabilization of validation loss after epoch 4 indicates strong generalization. Improved axis labels and annotations highlight key trends

Segmentation Results

Figure 8 showcases the segmentation results, highlighting the progression from raw MRI images to the final segmented regions. Each step reveals the model's ability to accurately delineate glioma areas.

Performance Analysis

The model's performance was evaluated using metrics such as F1 score, precision, recall, and Dice coefficient [13]. These metrics provide a comprehensive understanding of the model's ability to segment glioma regions from T1-weighted MRI images. The use of Dice Loss during training was particularly effective, as it directly measures the overlap between predicted segmentation masks and ground truth regions, ensuring accurate and consistent segmentation performance [8]. The results demonstrate the model's robustness and accuracy, especially in delineating glioma regions from healthy brain tissue.

While a confusion matrix is commonly used to provide pixel-level classification insights, its use is less critical in this context due to the focus on segmentation metrics like Dice coefficient and F1 score [13]. These metrics inherently capture the model's ability to handle class imbalances and evaluate segmentation quality effectively. The reliance on Dice Loss and related metrics ensures that the evaluation process aligns with the specific requirements of medical imaging tasks, where precise boundary delineation is crucial.

Future enhancements could include generating a confusion matrix to complement the current analysis by providing additional insights into pixel-wise misclassifications. This could be particularly beneficial for understanding the model's performance in distinguishing between tumor and non-tumor regions more granularly.

Discussion

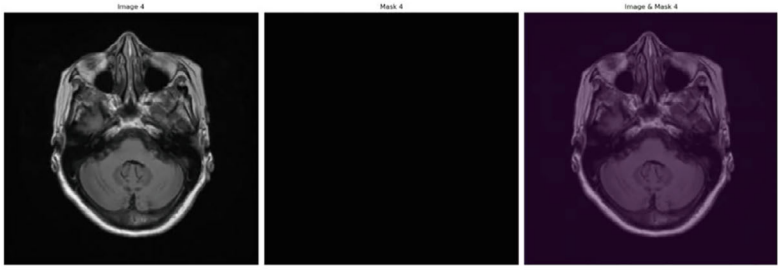
The results indicate that the model successfully segmented gliomas in T1-weighted MRI images, achieving high accuracy and F1 scores during training. The nn-UNet and VGG-UNet variants demonstrated complementary strengths:

- **nn-UNet**: Its automated configurations enhanced segmentation accuracy, particularly in distinguishing between low- and high-grade gliomas.
- **VGG-UNet**: The deeper architecture and hierarchical feature extraction capabilities improved segmentation precision.

The extensions to nn-UNet, such as deep supervision and normalization adjustments based on patient data, significantly enhanced the segmentation results. Moreover,

(a) (1) Original MRI image with detected glioma regions.

(b) (2) Binary segmentation mask highlighting glioma regions.



(c) (3) MRI image without detected glioma regions, demonstrating a false negative scenario.

(d) (4) Comparison of segmentation predictions with ground truth, showing accurate glioma detection.

Fig. 8 Additional segmentation results showing various steps of the segmentation process. Subfigure (3) highlights a false negative case, where no glioma region was detected, while the other subfigures illustrate successful detections. This comparison underscores the model's strengths and limitations in glioma segmentation

Table 2 Comparative performance analysis of segmentation models

Model	Test loss	Test accuracy	Test F1 score	Test precision	Test recall
Proposed U-Net	0.2149	0.9971	0.8618	0.8656	0.8581
VGG-UNet for Brain Tumor Segmentation [11]	0.2013	0.9985	0.8794	0.8821	0.8762
Extending nn-UNet [12]	0.2087	0.9969	0.8702	0.8740	0.8665

the VGG-UNet's feature extraction further refined the segmentation of subtle glioma features, crucial for identifying complex tumor boundaries.

Comparison with Existing Techniques

The proposed approach was compared with state-of-the-art models from recent literature:

- The **VGG-UNet for Brain Tumor Segmentation and Survival Prediction** achieved slightly higher F1 scores, primarily due to the use of an ensemble modeling approach. This method effectively mitigates biases inherent to individual models, resulting in improved robustness and accuracy in tumor segmentation and survival prediction tasks [11].
- Similarly, the **Extended nn-UNet for Brain Tumor Segmentation** demonstrated competitive results by incorporating advanced architectural adjustments. These enhancements contributed to superior segmentation performance, particularly in distinguishing between low-grade and high-grade gliomas [12].

Table 2 presents a detailed comparative analysis of the performance metrics, highlighting the strengths and differences between the proposed model and these benchmark approaches. This evaluation underscores the importance of architectural choices and advanced modeling techniques in achieving optimal segmentation outcomes.

Conclusions

This study successfully demonstrated the potential of leveraging advanced deep learning architectures for glioma segmentation in T1-weighted MRI images. The combination of nn-UNet and VGG-UNet provided robust segmentation results, showcasing complementary strengths in feature extraction and automated configuration. Metrics such as F1 score, precision, recall, and Dice coefficient highlighted the effectiveness of the proposed models in accurately identifying glioma regions.

These results underscore the capability of these architectures to assist in medical diagnostics by automating segmentation processes, saving time, and providing consistent results.

Limitations and Future Directions

Despite the promising outcomes, this study encountered several limitations that warrant further research:

- **GPU Dependence**: The heavy reliance on GPU resources restricted the scalability of training to larger datasets, limiting the breadth of experimentation.
- **Computational Requirements**: The increased model size and computational demands of the VGG-UNet and nn-UNet architectures reduced their feasibility for deployment in resource-constrained settings, such as smaller clinics or facilities with limited computing power.
- **Low-Grade Glioma Segmentation**: Differentiating low-grade gliomas from surrounding tissues posed significant challenges due to their subtle intensity differences, leading to occasional misclassifications.

Looking forward, several strategies could address these limitations and improve future segmentation models:

- **Transformer-based Architectures**: Combining convolutional layers with attention mechanisms, as in transformer models, could further enhance segmentation accuracy by capturing complex spatial relationships within MRI images.
- **Federated Learning**: Training models across multiple institutions without sharing sensitive patient data could enhance model generalizability while preserving privacy and adhering to ethical considerations.
- **Semi-supervised Learning**: Leveraging both labeled and unlabeled data could improve performance in settings where annotated datasets are scarce, which is often the case in medical imaging.

References

1. Jason Wang, Luis Perez, et al. “The effectiveness of data augmentation in image classification using deep learning”. In: *Convolutional Neural Networks Vis. Recognit* 11.2017 (2017), pp. 1–8.
2. Hongming Shan et al. “Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction”. In: *Nature Machine Intelligence* 1.6 (2019), pp. 269–276.
3. Mateusz Buda et al. *LGG MRI Segmentation Dataset of Brain Tumors*. Version 1.0. Associated with paper: “Systematic evaluation of image augmentation for deep learning-based brain tumor segmentation”, published in Artificial Intelligence in Medicine. Kaggle, 2019. DOI:

10.1016/j.artmed.2019.101746. URL: https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation.

4. Spyridon Bakas et al. “Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge”. In: *arXiv preprint* arXiv:1811.02629 (2018).
5. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.
6. Fabian Isensee et al. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211.
7. Md Abdullah Al Nasim et al. “Brain tumor segmentation using enhanced u-net model with empirical analysis”. In: *2022 25th International Conference on Computer and Information Technology (ICCIT)*. IEEE. 2022, pp. 1027–1032.
8. Thien B Nguyen-Tat et al. “Enhancing brain tumor segmentation in MRI images: A hybrid approach using UNet, attention mechanisms, and transformers”. In: *Egyptian Informatics Journal* 27 (2024), p. 100528.
9. Jieneng Chen et al. “Transunet: Transformers make strong encoders for medical image segmentation”. In: *arXiv preprint* arXiv:2102.04306 (2021).
10. Konstantinos Kamnitsas et al. “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation”. In: *Medical image analysis* 36 (2017), pp. 61–78.
11. Ali Nawaz et al. “VGG-UNET for brain tumor segmentation and ensemble model for survival prediction”. In: *2021 International Conference on Robotics and Automation in Industry (ICRAI)*. IEEE. 2021, pp. 1–6.
12. Huan Minh Luu and Sung-Hong Park. “Extending nn-UNet for brain tumor segmentation”. In: *International MICCAI brainlesion workshop*. Springer. 2021, pp. 173–186.
13. Mohammad Amin Habibi et al. “The performance of machine learning for prediction of H3K27 M mutation in midline gliomas: a systematic review and meta-analysis”. In: *World Neurosurgery* 186 (2024), e7–e19.

Vision Transformers for the Segmentation of Patterns Associated to Pathology in Diabetic Retinopathy: A Literature Review and Clinical Relevance in Eye Fundus Images

B. L. López-Covarrubias, L. J. González-Zazueta, J. I. Nieto-Hipolito, and G. J. Avilés-Rodríguez

Abstract Transformer-based models significantly enhance diabetic retinopathy detection and classification compared to traditional methods. However, challenges remain, particularly the need for large labeled datasets and the difficulty of interpreting model results. If integrated into clinical practice, these models could transform diabetic retinopathy diagnosis, enabling earlier, more precise interventions, reducing diagnostic costs, and increasing access to quality visual health care. Overall, while promising, further research is needed to optimize these models for clinical use, particularly in balancing performance with computational demands. A systematic review of studies from 2018 to 2024, following PRISMA guidelines, was conducted to assess the accuracy of these models, efficiency, and clinical relevance. The main findings of this research are: the integration of advanced models such as transformers could facilitate clinical decision-making, optimizing the time and resources of healthcare professionals, but while transformers can offer accurate predictions, they often fail to provide sufficient insight into how those predictions were reached, making clin-

These authors contributed equally to this work.

B. L. López-Covarrubias · L. J. González-Zazueta
Maestría y Doctorado en Ciencias e Ingeniería (MyDCI), Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California, Ensenada, BC, Mexico
e-mail: lariza.lopez@uabc.edu.mx

L. J. González-Zazueta
e-mail: lgonzalez85@uabc.edu.mx

J. I. Nieto-Hipolito
Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California, Ensenada, BC, Mexico
e-mail: jnieto@uabc.edu.mx

G. J. Avilés-Rodríguez (✉)
Escuela de Ciencias de la Salud, Universidad Autónoma de Baja California, Ensenada, BC, Mexico
e-mail: aviles.gener@uabc.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_5

ical adoption difficult, this opens an opportunity for the development of intelligent computing models with capabilities to show the how the output of the algorithm is generated.

Keywords Vision transformers · Diabetic retinopathy · Digital fundoscopy · Computational ophthalmology · Computational medicine

1 Introduction

Diabetic retinopathy (DR) is a common ocular complication in individuals with Diabetes Mellitus, which can lead to blindness if not detected and treated timely. Globally, DR affects millions of people, significantly impacting their quality of life and increasing the burden on healthcare systems. Therefore, early detection and treatment planning are essential to prevent irreversible damage. The International Clinical Diabetic Retinopathy Disease Severity Scale (ICDR) is the standard for defining the level of damage caused by DR [1]. Fundus images are key tools for identifying and assigning a severity level, and their interpretation is performed by visual health specialists. However, there are methods that can optimize this process [2].

In recent years, advances in artificial intelligence have revolutionized medical diagnosis, with convolutional neural networks (CNN) standing out. These have been fundamental in automated diagnosis using medical images, although they present limitations by not capturing long-range dependencies. In 2017, the transformer architecture was introduced for natural language processing [3], and in 2020, Dosovitskiy and colleges [4] presented the Vision Transformer (ViT) for image processing. Unlike CNNs, transformer-based models, such as ViT, can capture global dependencies in images by dividing them into non-overlapping instances. Although there are fewer studies on ViT compared to CNNs, they are considered promising for diagnosis using medical images.

This study reviews and analyzes recent literature on the classification and detection of DR using models that integrate transformers. Understanding the complexity of these models is crucial for improving accuracy and efficiency in diagnosis, benefiting patient care. Our paper aims to expand the current understanding of DR classification methodologies, identify key trends, and highlight unresolved challenges. We conducted a literature analysis from 2018 to 2024, following the PRISMA guidelines to ensure a rigorous collection and critical analysis. The potential impact of adopting these technologies in clinical practice is significant. It could reduce the costs associated with diagnosis, increase accessibility to quality visual health services, and improve patient health outcomes by enabling earlier and more accurate interventions. Furthermore, the integration of advanced models such as transformers could facilitate clinical decision-making, optimizing the time and resources of healthcare professionals.

2 Methods

This work explores the literature in the field of DR classification using transformer-based algorithms. We followed the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [5] to ensure a comprehensive review.

2.1 Search Strategy

The article search was conducted through PubMed, IEEE Xplore, SCOPUS, and Google Scholar, using a combination of keywords as shown in Table 1. Additionally, the search was limited to English-language journal articles. PubMed yielded 245 results, IEEE 110, SCOPUS 127, and for Google Scholar, we selected articles added during the last year, sorted by date and abstract search, obtaining 76 results. Table 1 shows the specific queries used in each of the databases consulted and the initial amount of publications located. Whereas Fig. 1 shows the process through which we selected the articles used for this survey, following the PRISMA methodology as mentioned in the previous parragraph.

2.2 Article Selection

The goal in this step was to perform a systematic search to identify relevant studies that implement the transformer architecture in the classification of DR. After searching the databases, a total of 558 articles were obtained. From these, duplicates were removed, and a preliminary review was conducted to determine eligibility based on title and abstract. Articles were excluded based on the following criteria: studies with images of animals or premature infants, case studies, trials, reviews, and bibliometric analyses, models not related to transformers, evaluations and comparisons between models, data augmentation and image quality improvement, multiple diseases or not related to diabetic retinopathy, lesion or anatomical region segmentation and classification, image types from techniques outside of digital fundoscopy, and unrelated topics.

For the full-text review, articles were excluded if they proposed algorithms exclusively for binary classification (presence or absence of DR), focused solely on identifying lesions without assigning a severity classification for DR, evaluated algorithms targeting specific areas of improvement, or could not be retrieved.

The included articles are based on studies that propose algorithms whose architecture includes transformers and their variants. Additionally, they meet the following criteria: use of fundus images, classification of DR according to its severity, original research articles, studies that include quantitative metrics to evaluate performance,

Table 1 Search queries and initial amount of publications located per database

Database	Search query	Results
PubMed	(“Vision Transformers” OR “Transformer*” OR “ViT” OR “Self-Attention” OR “Attention*”) AND (“Diabetic Retinopathy” OR “DR” OR “Segment*” OR “Image Segmentation” OR “Detect*” OR “Classif*” OR “Localizat*” OR “Grad*”) AND (“fundoscop*” OR “Fundus Image*” OR “Retinal Image*” OR “Retinal Photography” OR “Ocular Fundus Photography”) AND English[lang] AND “journal article”[Publication Type] NOT (“review”[Title] OR “systematic review”[Title] OR “meta-analysis”[Title])	245
IEEE Xplore	(“Vision Transformers” OR “Transformer*” OR “ViT” OR “Self-Attention” OR “Attention*”) AND (“Diabetic Retinopathy” OR “DR” OR “Segment*” OR “Image Segmentation” OR “Detect*” OR “Classif*” OR “Localizat*” OR “Grad*”) AND (“fundoscopy” OR “Fundus Image*” OR “Retinal Image*” OR “Retinal Photography” OR “Ocular Fundus Photography”) NOT (“review” OR “systematic review” OR “meta-analysis”)	110
SCOPUS	(TITLE-ABS-KEY (“Vision Transformers” OR “Transformer*” OR “ViT” OR “Self-Attention” OR “Attention*”)) AND (TITLE-ABS-KEY (“Diabetic Retinopathy” OR “DR” OR “Segment*” OR “Image Segmentation” OR “Detect*” OR “Classif*” OR “Localitat*” OR “Grad*”)) AND (TITLE-ABS-KEY (“fundoscop*” OR “Fundus Image*” OR“Retinal Image*” OR “Retinal Photography” OR “Ocular Fundus Photography”)) AND PUBYEAR > 2017 AND PUBYEAR < 2025 AND (LIMIT-TO (SRCTYPE, “j”)) AND (LIMIT-TO (DOCTYPE,“ar”)) AND (LIMIT-TO (SUBJAREA,“COMP”)) AND (LIMIT-TO (LANGUAGE, “English”)) AND (LIMIT-TO (EXACTKEYWORD,“Attention Mechanisms”) OR LIMIT-TO (EXACTKEYWORD,“Attention”) OR LIMIT-TO (EXACTKEYWORD,“Attention Mechanism”) OR LIMIT-TO (EXACTKEYWORD,“Vision Transformer”) OR LIMIT-TO (EXACTKEYWORD,“Attention Network”) OR LIMIT-TO (EXACTKEYWORD,“Transformer”) OR LIMIT-TO (EXACTKEYWORD,“Transformers”))	127
Google Scholar	“Diabetic Retinopathy” AND (“Vision Transformer” OR “ViT” OR “Transformer*” OR “Swin Transformer” OR “Self-Supervised Transformer” OR “Conv-ViT”) AND (“Classification” OR “Detection” OR “Grading” OR “Prediction” OR “Severity”)	76
Total		558

and studies that propose substantial innovations in transformer-based model architectures, such as new preprocessing strategies, classification techniques, or hybrid approaches.

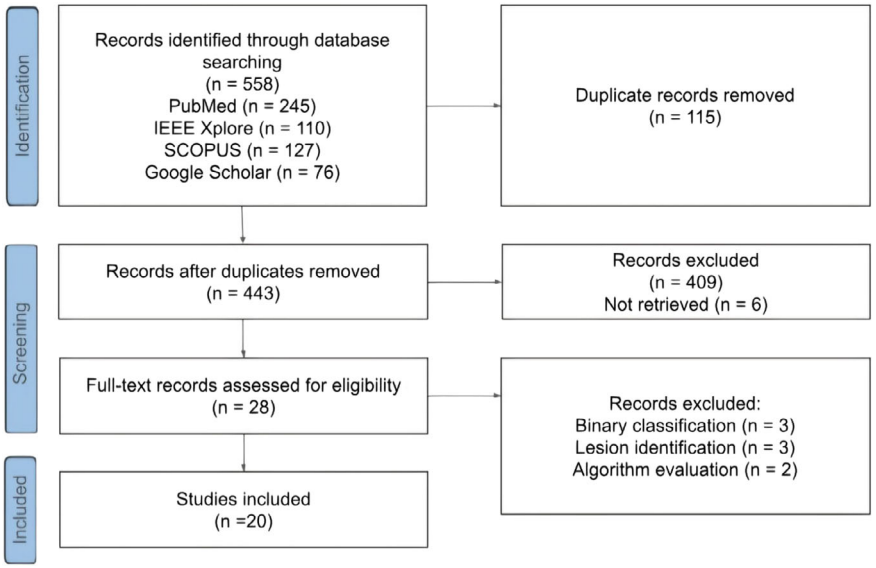


Fig. 1 Flow diagram for article selection of this work, following the PRISMA methodology [5]

We obtained a total of 558 articles from the 4 databases, removed 115 duplicates, and excluded 409 based on title and abstract according to the exclusion criteria. Six articles were not retrieved, leaving 28 for full-text review. Finally, for this study, we included 20 articles, comprising 4 conference papers, 15 journal publications, and one preprint article. The overall article selection process is shown in Fig. 1.

3 Results

Of the 20 selected articles, 10 implement hybrid techniques, most of which integrate CNNs, widely used in the segmentation and/or classification of medical images. Additionally, one study was found that integrates ViT and a Capsule Network (CapsNet). CapsNet [6], was first proposed by Sabour and colleagues and addresses issues in CNNs, such as the loss of information during pooling operations. Lastly, a model incorporating Class-Specific Residual Attention (CSRA) is included. The other 10 articles are based on non-hybrid architectures, relying entirely on transformer derivatives such as ViT and Swin Transformers. Figure 2 shows a schematic classification of the general concepts presented in the 20 articles selected.

The articles included in this study aim to classify DR through models that integrate transformer models in their architecture. In all cases, the classification is based on severity. Additionally, some complement this approach with binary classification [7–11], distinguishing the presence or absence of DR, while others also seek the

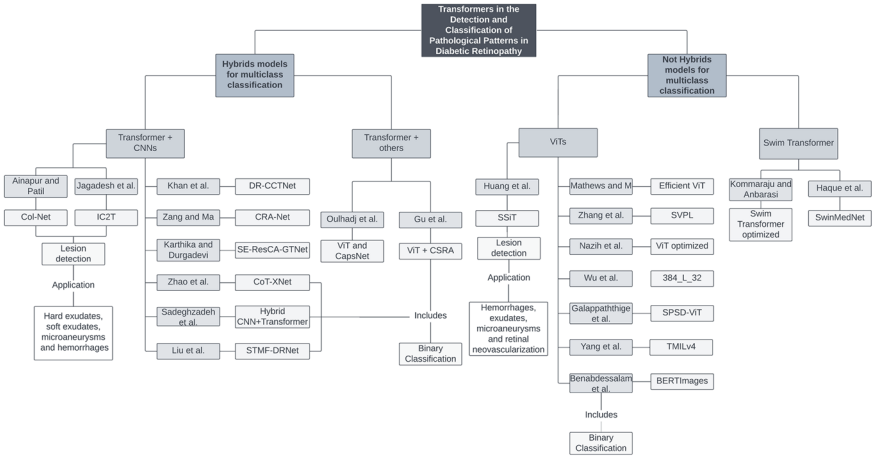


Fig. 2 Taxonomy of concepts derived from subset of articles selected for this work

detection of specific lesions related to DR such as microaneurysms, hemorrhages, and exudates to improve diagnostic accuracy [12–14].

The authors classify DR based on its severity, using five levels of severity: Class 0 (No DR), Class 1 (Mild Non-Proliferative DR), Class 2 (Moderate Non-Proliferative DR), Class 3 (Severe Non-Proliferative DR), and Class 4 (Proliferative DR). In some studies, such as [8, 15], an additional class identified as "Unclassifiable" is included, for images that do not meet the quality standards required for training.

The following section describes the models proposed by the articles in this review, highlighting the key elements and their main function for the task.

3.1 Hybrid Algorithms

3.1.1 Transformers and CNNs

This section approaches publications where a combination of transformers and CNNs are combined to attack the problem of DR classification or segmentation. Table 2 shows a summary of the most relevant findings in this subset of articles, as well as the databases and metrics corresponding to multiclass classifications, along with general information about the computational resources used and relevant data on computational efficiency, when provided by the authors.

The abbreviations used in Table 2 can be interpreted as follows: Accuracy (Acc), Precision (Prec), Specificity (Spec), Sensitivity (Recall), F1 Score (F1), Quadratic Weighted Kappa (QWK), Area Under the Curve (AUC), Intersection over Union (IoU), Balanced Accuracy (Balanced Acc), Kappa Score (Kappa), and Top-1 Accuracy (Top-1 Acc).

Table 2 Summary of transformers and CNNs models

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
DR-CCTNet	Compact Convolutional Transformer	APTOS, Messidor-2, IDRiD, DDR, Kaggle Diabetic Retinopathy Acc: 90.17%, Prec: 89.38%, Spec: 97.51%, F1: 89.72%	Training time of 16 s per epoch
CRA-Net	DenseNet121, Category attention block, Lesion recognition module	APTOS Acc: 89.10%, QWK: 0.932 DDR Acc: 83.10%, QWK: 0.840	NVIDIA GTX 3090TI
SE-ResCA-GTNet	D2_UNet, ResNet_101, Squeeze Excited block, Global attention mechanism	Kaggle EyePACS and 40 sets of images Acc: 99.8%, Prec: 99.2%, Spec: 99.7%, Recall: 99.4%, F1: 99.3%, Kappa: 0.991	–
CoT-XNet	Contextual transformer, Xception network-based on CNNs	APTOS Acc: 84.18%, Kappa: 0.9000 DDR Acc: 83.01%, Kappa: 0.8496 EyePACS Acc: 84.10%, Kappa: 0.7684	AMD Ryzen 9 5950x, 128 GB RAM, 2 NVIDIA RTX 3090 GPUs with 12 GB of memory
Hybrid CNN+Transformer	EfficientNet-B0, Transformer encoder-decoder, Multi-Head Self Attentions, Residual Spatial Module, Feed-forward network	APTOS Acc: 94.47%, QWK: 0.8136 EyePACS Acc: 95.93%, QWK: 0.8956 DDR Acc: 92.56%, QWK: 0.8308 Messidor-2 Acc: 98.21%, QWK: 0.8963	Intel Core i9 CPU with a 3.60 GHz processor, 64 GB of RAM, NVIDIA GeForce RTX 2070 GPU with 8 GB of memory
STMF-DRNet	Swin Swin Transformer V2 is the main architecture, Multi-branch network Convolutional, Block Attention Module	APTOS Acc: 85.50%, F1: 85.90%, Kappa: 0.872 EyePACS Acc: 86.30%, F1: 87.70%, Kappa: 0.841 DDR Acc: 87.90%, F1: 89.10%, Kappa: 0.890	CentOS 7.8, NVIDIA GeForce GTX 3060Ti GPU with 12 GB of memory

(continued)

Table 2 (continued)

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
DR-CCTNet	Compact Convolutional Transformer	APTOS, Messidor-2, IDRiD, DDR, Kaggle Diabetic Retinopathy Acc: 90.17%, Prec: 89.38%, Spec: 97.51%, F1: 89.72%	Training time of 16 s per epoch
Col-Net	VGG16_bn, Conjugated attention mechanism, ViT, F-Net, Kernel Extreme Learning Machine	Combination of: APTOS, Messidor-2, DIARETDB1 V2.1, DIARETFB0 Acc: 99.63%, Prec: 99.87%, Spec: 99.12%, Sens: 99.52%, F1: 98.76%, IoU: 98.26%	Test time of 101.12 s, using an NVIDIA Tesla K40 GPU with 2800 CUDA cores.
IC2T	Optical disc detection module, Blood vessel detection module, Dual convolutional transformer	Combination of: EyePACS, Messidor-2, DIARETDB0. For the entire dataset by class (1–4): Acc: 98.08% to 99.75%, Prec: 0.93 to 1.00%, Recall: 0.96 to 0.99%, F1: 0.96 to 0.99%	Intel Xeon E5-2620 CPU at 2.4 GHz, 3 NVIDIA GPUs, each with 12 GB of RAM

Khan and colleagues [16] compared 4 transfer learning models: VGG16, VGG19, MobileNetV2, ResNet50, and 3 transformer-based models: Vision Transformer (ViT), Swin Transformer, and Compact Convolutional Transformer (CCT). Using CCT as the base model, they proposed DR-CCTNet. Twelve ablation studies were implemented to find an optimal configuration in order to reduce computational complexity and training time. Compared to the original CCT, DR-CCTNet has a transformer encoder block, making it less complex. A total of 53,185 raw images were used from the following datasets: APTOS, Messidor2, IDRiD, DDR, and Kaggle Diabetic Retinopathy, resulting in a combined dataset of 154,882 images with data augmentation for DR classification. The model outperformed the others with a test accuracy of 90.17%, even with low-resolution images and a training time of 38 s per epoch.

Zang and Ma [17] address the problem of class imbalance and the relationships between lesions for the classification of each class by proposing CRA-Net. After a CNN (DenseNet121) extracts local features, a category attention block and a lesion attention block guided by a transformer come into play. The latter contains two capsules: a capsule attention to identify and model the most representative lesions,

and a spatial attention, which, through a heatmap generated by the transformer, allows the localization of the most representative lesions.

Zhao and colleagues [8] propose CoT-XNet, which combines a Contextual Transformer (CoT) and an Xception network based on CNNs. The 3 x 3 convolutional layers of the original SE-ResNet-152 model are replaced to introduce the CoT module. Thus, the images are processed through two parallel paths to extract features separately, which are then concatenated into a 4096-dimensional vector for final prediction through fully connected layers.

Sadeghzadeh [9] integrates three main modules: EfficientNet-B0 for feature extraction, a Transformer Encoder-Decoder (TE-TD) which, along with Multi-Head Self Attentions (MHSA), captures and encodes global features, and a feed-forward network (PFFN) for classification. Additionally, a Residual Spatial Module (RSM) is integrated, which receives the outputs from the transformer, optimizing performance.

SE-ResCA-GTNet by Karthika and Durgadevi [18] uses $D2_U Net$ for lesion segmentation, $ResNet_101$ (pre-trained on ImageNet) for feature extraction, a Squeeze Excited (SE) block to focus on key features, a global transformer block to capture complex relationships, and Cross-field Attention (CA) to enhance the identification of important features for DR. Additionally, Gazelle Optimization (GO) is integrated as an optimization algorithm to improve final accuracy, along with preprocessing techniques that include Equalized-Joint Histogram (E-Join Hist). SE-ResCA-GTNet primarily focuses on segmenting blood vessels, leveraging this for subsequent real-time DR classification. Despite achieving high performance metrics, it requires long training times.

A model called STMF-DRNet is proposed, combining Swin Transformer V2 with a multi-branch network for multi-level feature extraction and a Convolutional Block Attention Module (CBAM) to enhance feature extraction. This architecture, developed by Liu [10], complements the base transformer by integrating these specialized modules to improve performance.

Ainapur and Patil [12] propose Col-Net, which is based on $VGG16\_bn$ for initial feature extraction, integrating a conjugated attention mechanism to highlight important areas and a ViT to capture long-range relationships in the images. Additionally, the model includes F-Net to fuse features from multiple images and Kernel Extreme Learning Machine (KELM) to identify DR cases according to severity. The model also detects lesions such as hemorrhages, microaneurysms, soft exudates, and hard exudates. For the DIARETDB1 V2.1 dataset, the highest classification accuracy was achieved for hard exudates at 99.25%, while the highest error rate was for hemorrhage cases at 1.87%, and 2.24% of microaneurysms were mistakenly identified as hemorrhages.

Jagadesh [13] enhance a Contour Convolutional Transformer (IC2T) by incorporating an Optical Disc (OD) detection module and a Blood Vessel (BV) detection module to improve the model's focus on critical regions. Additionally, a dual convolutional transformer block (DC) is integrated to establish connections between local and global features. To address the loss of detail when reducing image size, bicubic interpolation is used, which calculates new pixel values based on neighboring pixel values. The authors also report lesion detection and DR severity classification based

on those lesions, with a test time of 101.12 s. In the DIARETDB1 dataset, the lesions with the highest classification errors were hemorrhages (H.M.) and microaneurysms, with an F1-Score of 0.72 and an accuracy of 0.87, although they achieved a high recall rate of 0.99.

3.1.2 Transformers and Other Algorithms

Table 3 in this section provides a summary of the key elements of the hybrid models that combine Transformers with other models not based on CNNs, using the same abbreviations for key metrics from Table 2.

Oulhadj [19] combined Vision Transformer (ViT) and Capsule Network (CapsNet) to manage the spatial relationship between pixels, avoiding making the model too deep in order to reduce complexity. Additionally, a novel preprocessing composition is introduced, which includes Power Law Transformation (PLT) and CLAHE (applied only to the input images for CapsNet). The resized images are fed into ViT (224,224,3) with a patch size of 16 x 16, and into CapsNet (120,120,1). ViT was not trained from scratch, as a pre-trained model was fine-tuned to the dataset used. The model achieved a training time per epoch between 3.25 and 16 min, with preprocessing and prediction times per image ranging from 0.6 to 1.1 s, depending on the dataset. It is noteworthy that the authors implemented cross-dataset training and testing, resulting in lower accuracy compared to evaluations without cross-dataset validation.

On the other hand, Gu [15] combine ViT and Class-Specific Residual Attention (CSRA) in a model composed of two main modules: the Feature Extraction Block (FEB) and the Grade Prediction Block (GPB) for classification prediction. FEB operates under the transformer architecture to capture relationships between image patches, while GPB uses CSRA as part of its mechanism to enhance DR classification by applying attention to key features for each class. Additionally, binary cross-entropy loss is included to calculate the error between predictions and actual labels, and the gradient descent method is used to optimize the loss function.

3.2 Non-hybrid Algorithms

3.2.1 Vision Transformers

Table 4 in this section summarizes the key elements of non-hybrid ViT models, using the same abbreviations for key metrics from Table 2.

Mathews [20] optimize a model based on the ViT architecture with an enhanced multi-headed self-attention (MSA) mechanism. While ViT can be computationally expensive, the authors address this issue by introducing the Linformer technique, which reduces the quadratic complexity of self-attention to linear. In this way, they propose Efficient ViT, achieving a batch latency time of 0.00832 s.

Table 3 Summary of transformers and other models

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
ViT and CapsNet	ViT, Capsule network	APTOS Acc: 88.18%, Prec: 80.00%, Sens: 76.00%, F1: 78.00% Messidor-2 Acc: 87.78%, Prec: 87.00%, Sens: 87.00%, F1: 87.00% DDR Acc: 80.36%, Prec: 59.00%, Sens: 55.00%, F1: 56.00% EyePACS Acc: 78.64%, Prec: 54.00%, Sens: 42.00%, F1: 46.00%	Training time per epoch: 3.25 min (APTOS), 1.8 min (Messidor-2), 11.66 min (DDR), 16 min (EyePACS)
ViT + CSRA	Feature extraction block, Grade prediction block	Reported metrics by class: DDR Acc: 77.93 to 97.52%, Prec: 63.39 to 80.05%, Spec: 0.57 to 21.59%, F1: 51.30 to 86.44% IDRiD Acc: 72.82 to 98.06%, Prec: 53.70 to 100%, Spec: 2.00 to 18.18%, F1: 66.67 to 86.96%	NVIDIA Quadro RTX 6000 GPU with 24 GB of memory

Zhang [21] propose Semantic-oriented Visual Prompt Learning (SVPL), a model that learns DR classification knowledge from large-scale pre-trained models (LPM). ViT serves as the base architecture, dividing the images and processing them through transformer encoder layers. SVPL consists of three main modules: Visual Prompt Grouping (VPG), Contrastive Group Alignment (CGA), and Hierarchical Semantic Delivery (HSD). The model is trained using mixed cross-entropy loss (MCE) to enhance performance, achieving an average inference time of 0.54 s.

Nazih [22] optimize a ViT model by incorporating an advanced optimizer (AdamW) and applying regularization techniques such as dropout and L2 regularization. They also address the issue of data imbalance in the FGADR dataset, which

Table 4 Summary of ViT models

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
Efficient ViT	ViT with an enhanced multi-headed self-attention mechanism	IDRiD Acc: 88.13%, Prec: 88.52%, Recall: 88.13%, F1: 88.07%	The average latency per batch was 0.00832 s, using an NVIDIA GeForce RTX 3060 con 12 GB de RAM.
SVPL	Visual Prompt Grouping, Contrastive Group Alignment, Hierarchical Semantic Delivery	EyePACS, APTOS, DDR QWK: 0.892	Average inference time of 0.54 s, GPU NVIDIA Quadro RTX 6000
ViT optimized	Based on ViT	FGADR F1: 82.5%, Acc: 82.5%, Balanced Acc: 82.6%, AUC: 0.964, Prec: 82.5%, Recall: 82.5%, Spec: 95.6%	Tesla V100 with 16 GB of memory and 12 GB of RAM
384L32	Based on ViT	Kaggle Acc: 91.4%, Spec: 97.7%, Prec: 92.8%, Recall: 92.6%, QWK: 0.935, AUC: 0.986	Intel(R) Core(TM) i9-10920X CPU at 3.50 GHz with 32.0 GB of RAM, using an NVIDIA GeForce GTX 3080 10G GPU
SPSD-ViT	Architecture based on ViTs with a focus on domain generalization	APTOS, EyePACS, Messidor, Messidor-2 Multi-source domain generalization Average top-1 Acc: 65.5% Single-source domain generalization Average top-1 Acc: 55.9%	Average step time of 0.368 s per step, 4 NVIDIA RTX A6000 GPUs
SSiT	The model is trained using two main methods: A contrastive loss guided by saliency maps and a saliency map segmentation loss.	APTOS Fine-tuning: 92.97% k Linear evaluation: 89.65% k k-NN classification: 84.65% k Messidor-2 Fine-tuning: 77.53% k Linear evaluation: 67.23% k k-NN classification: 49.42% k DDR Fine-tuning: 81.88% k Linear evaluation: 71.33% k k-NN classification: 58.89% k	4 GPUs NVIDIA RTX 3090

(continued)

Table 4 (continued)

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
TMILv4	ViT-Small, Global Instance Computing Block	APTOS Acc: 85.6±1.4%, Sen: 73.7±1.6%, AUC: 95.6±0.7%, Kappa: 90.7±0.6	TMIL achieved a 62% reduced inference time compared to the original ViT, using an NVIDIA RTX 3090 GPU with 24 GB of VRAM
BERTImages	Multi-head attention mechanisms (8 heads)	Kaggle Prec: 96.8%, Recall: 96.7%, F1: 96.6%	–

contains high-resolution images. To mitigate this, they performed data augmentation, class weighting, and label smoothing to prevent the model from becoming overly confident in its predictions, along with the use of focal loss. The authors highlight the computational demands of this model compared to other architectures like CNN, but justify it with the results, achieving a precision and sensitivity of 0.85 for both metrics.

Wu [23] aim to demonstrate that transformer-based architectures alone are suitable for DR classification. They conduct a series of tests comparing different configurations, with the standout being 384_L_32, which uses the "Large" version of the ViT model, implying more layers and parameters.

Galappaththige [24] develop SPSD-ViT, an architecture based on ViTs with a focus on domain generalization (DG). They seek to enhance the algorithm with self-distillation, allowing the model to learn from its own predictions during training. Additionally, they integrate an adaptive prediction smoothing mechanism, which prevents the model from making overly confident predictions by avoiding assigning a very high probability to one class while drastically lowering probabilities for others. This results in a more flexible model. SPSD-ViT outperformed SD-ViT, which also uses self-distillation but with a fixed smoothing mechanism, and ERM-ViT, which only minimizes empirical loss. Furthermore, it achieved a time of 0.368 s per step, which is only slightly higher than other methods.

Huang [14] developed SSiT, a self-supervised ViT that leverages saliency maps for guidance during self-supervised training. The model is trained using two main methods: a saliency map-guided contrastive loss and a saliency map segmentation loss. The contrastive loss helps the model focus on relevant areas for classification through saliency maps, while the segmentation loss enables the model to learn to segment features, such as lesions or blood vessels, and ultimately classify them.

Yang [7] propose a transformer-based model with multiple instance learning (TMIL). They conducted an ablation study comparing different ViT models: image

sizes of 224 x 224 and 896 x 896, and four TMIL configurations: with and without a Global Instance Computing Block (GICB), with positional embeddings, and with residual connections combined with GICB. The final configuration (TMILv4) was identified as the best to continue the study. After the ablation study, each image was divided into 16 instances of 224 x 224 while maintaining the original resolution of 896 x 896. ViT-Small was used to extract features from each instance. Then, the GICB, equipped with multi-headed self-attention mechanisms, captured relationships between features and global dependencies. Notably, TMIL achieved a 62% reduction in inference time compared to the original ViT model.

Benabdessalam [11] propose BERTImages, a modified version of Bidirectional Encoder Representations from Transformers (BERT), which has shown high performance in natural language processing (NLP) tasks such as question answering, sentiment analysis, and translation. BERT operates similarly to ViT, utilizing transformer blocks but with words instead of images. In the proposed model, the transformer architecture is adopted, with image instances of 16 x 16 pixels (selected after several trials, achieving an accuracy of 0.90) processed through ResNet50. During the embedding phase, the instances are converted into vectors in a multidimensional space, capturing various features (pixels, shapes, and textures) while preserving spatial information. Multi-headed attention mechanisms (with 8 heads) are used to extract features, detecting long-range dependencies between instances.

3.2.2 Swin Transformer

Table 5 in this section summarizes the key elements of non-hybrid Swin Transformer models, using the same abbreviations for key metrics from Table 2.

Some authors emphasize preprocessing techniques. For instance, Kommaraju and Anbarasi [25] experiment with a Swin Transformer architecture, highlighting the importance of preprocessing by including CLAHE and Adaptive Median Filtering to reduce noise. Additionally, Haque [26] implement SwinMedNet, which is optimized for classifying the RetinaMNIST dataset subset, incorporating preprocessing techniques such as CLAHE, Multiscale Retinex (MSR), Gaussian filtering, and hyperparameter tuning.

4 Discussion

For years, CNNs have dominated the field of computer vision, especially in medical applications such as image segmentation using networks like U-Net [27]. Their ability to extract features allows them to achieve competitive accuracy in the classification and segmentation of medical images. One reason for their popularity is the property of translation invariance, which enables the use of the same kernels in various parts of the image, reducing parameters and simplifying training [22]. However, although convolutions can be parallelized, they face challenges in capturing

Table 5 Summary of swin transformer models

Model	Architectures and modules	Database and metrics	Resources and computational efficiency
Swin Transformer optimized	Based on Swin Transformer	Average of 10 runs APTOS: Acc: 94.01%, Spec: 93.89%, F1: 82.16%, Sens: 92.10%	NVIDIA GeForce GTX 1070 GPU in a laptop
SwinMedNet	Based on Swin Transformer	RetinaMNIST subset of the MedMNIST V2 dataset: Acc: 56.80%, AUC: 0.834	–

long-range relationships and detecting features at different scales [28]. Despite the historical effectiveness of CNNs in specific applications, transformer architectures are emerging as viable alternatives, especially in contexts where identifying complex relationships is crucial for achieving optimal precision. ViTs can process entire sequences in parallel. Their attention mechanism allows them to capture long-range dependencies and the global context of the image, assigning different levels of importance to various regions. This helps them focus on relevant areas, such as lesions in medical images, thereby improving accuracy in pathology identification. Additionally, unlike CNNs, they can handle images of different sizes [29], although their use entails greater computational complexity when calculating relationships among all elements of the input []. While transformers often generalize less effectively than CNNs when trained on small datasets, their performance improves significantly with large-scale pretraining, especially as their size and the amount of data used increase. When ViTs are trained on broad and diverse datasets, they demonstrate superior transfer capabilities and strong performance on complex tasks, such as image classification.

A transformer requires a large amount of data to reach its full potential, and its architecture based on attention mechanisms incurs high computational costs compared to CNNs, especially as the size of the image or the number of tokens increases. Collecting a large dataset is a challenge in medical imaging, where data may be limited, so it is often preferable to train them on large-scale datasets and then fine-tune them on a limited number of images, as in the case of diabetic retinopathy. This approach generates additional computational costs and has led some authors of the reviewed articles to propose hybrid models, mainly combining CNNs and transformers, to reduce computational complexity and processing requirements, which is a crucial challenge in clinical implementation. For example, the DR-CCTNet model, based on a compact transformer, reduced training time to 16 s per epoch and achieved an accuracy of 90.17%, surpassing models like ResNet and DenseNet,

which achieved lower results with the same dataset. For example, Khan [16]. On the other hand, Col-Net by Ainapur and Patil [12] consumed 101.12 s for testing, which was faster than other models in the study: 236.41, 277.60, and 197.12 s for CNN-SVM, CapsNet, MobileNet V3, respectively. Oulhadj [19] presents a hybrid model combining ViT and CapsNet, it had a training time per epoch between 3.25 and 16 min, with testing and prediction times per image ranging from 0.6 to 1.1 s depending on the dataset. However, hybrid algorithms do not always have lower computational costs. For example, SE-ResCA-GTNet [18], despite achieving superior performance metrics compared to other models, still has limitations, including long training times when working with large datasets. This could be due to the advanced mechanisms, such as cross-attention and Squeeze-and-Excitation, which require significant computations and parameter tuning.

Non-hybrid models, such as ViT and Swin Transformers can achieve promising results. Nazih [22] demonstrate that their model outperforms CNNs in terms of precision and sensitivity. However, due to its transformer-based nature, it requires higher computational power, making its practical implementation more challenging. Despite the known computational demands of ViT and Swin Transformer, efforts have been made to optimize training time. For instance, the transformer-based TMIL model, which incorporates multiple instances of learning, reduced inference time by 62% compared to the original ViT, representing a significant optimization in computational efficiency [7].

Regarding preprocessing techniques, CLAHE stands out as a widely used method among the reviewed articles. In fundus images, CLAHE enhances local contrast, highlighting important details such as microaneurysms, hemorrhages, and exudates. Gaussian blur is also commonly implemented to reduce noise, and data augmentation is frequently used to address the significant class imbalance often found in fundus image datasets.

Lesion identification plays a key role in determining the severity class of DR. Among the mentioned studies, only [12, 13] report classification metrics related to lesions, with exudates being the best-classified lesions. In contrast, hemorrhages and microaneurysms show lower classification efficiency, likely due to their smaller size, diffuse appearance, or lower contrast against the retinal background.

Most of the hybrid models addressed in this study incorporate elements derived from CNNs in their architecture, which makes sense given their ability to extract features such as edges, textures, and low-level patterns. In this review, CNNs are primarily used for initial local feature extraction [9, 17, 18].

Transformers can be integrated into models in various ways, offering new possibilities for improving medical image analysis. In the work of Khan, DR-CCTNet [16], the use of CCT incorporates convolutions as part of the initial process, enhancing local representations before the transformer models relationships globally. In contrast, Zang and Ma [17] take a different approach by integrating a transformer-guided lesion block, which is useful in scenarios where specific lesions need to be highlighted, such as microaneurysms, hemorrhages, or exudates, to enable accurate classification. Karthika and Durgadevi [18] and Jagadesh [13] use a transformer block to capture complex, long-range relationships, a task for which transformers are well-

suited. Zhao [8] combine CoT with an Xception network based on CNNs, a strategy that allows them to leverage the strengths of both CNNs and transformers in parallel, capturing global relationships and local details simultaneously. Sadeghzadeh [9] use a TE-TD with MHSA to capture and encode global features. Finally, Ainapur and Patil [12] integrate a ViT into their architecture to capture long-range relationships. The goal is to harness the benefits offered by transformers, capturing global dependencies across parts of fundus images and understanding long-range relationships, which can be challenging for architectures like CNNs alone.

The application of transformer-based models, such as ViTs, in the clinical field can lead to earlier and more accurate interventions, which is crucial for preventing vision loss in diabetic patients. Additionally, by facilitating image-based diagnosis, they have the potential to reduce medical costs and improve access to quality diagnostics, especially in areas with a shortage of specialists. However, a key challenge is that, although they can provide accurate results, it is not always clear how these conclusions are reached, which limits their clinical acceptance. Furthermore, their high computational cost can hinder real-time diagnostics, which is critical in clinical situations, such as diagnosing diabetic retinopathy, where quick decision-making is needed. While optimizations like hybrid models and multi-instance learning techniques have been implemented to reduce inference time [7], there is still a gap for these systems to operate in real time. Therefore, developing models that achieve a balance between diagnostic accuracy and computational efficiency is essential for their successful clinical implementation.

5 Conclusion

Through various approaches, researchers have implemented transformers in innovative ways, each with a specific focus on improving the representation and understanding of features in digital fundoscopy. These range from architectures based solely on ViT to specialized transformer blocks designed for lesion segmentation or feature extraction tasks. The authors of the reviewed articles show a clear preference for hybrid architectures, suggesting that there are significant approaches to consider for facilitating the application of artificial intelligence in the real world through these combined models. The most common architectures for DR classification are hybrid models combining transformers and CNNs, and Vision Transformer-based models. Hybrid models are effective because they capture local features using CNNs and global features (long-range dependencies) with transformers. Additionally, hybrid models tend to have lower computational costs. Using transformers, especially in their pure form, is computationally intensive, limiting their implementation in clinical environments with limited access to high-performance hardware. Continued development of lighter models that can operate efficiently without sacrificing performance is necessary.

The most common lesions segmented in diabetic retinopathy according to the results of this work include microaneurysms, hemorrhages, and hard and soft exu-

dates. These lesions are crucial for classifying DR into different stages of severity, enabling physicians to plan appropriate treatment. However, many models tend to overlook the importance of providing precise spatial locations of these lesions. Specialists prefer more detailed approaches when analyzing these images where lesions are clearly identified and localized, as this helps them make more informed diagnostic and treatment decisions. Without this level of detail, classification results tend to be less useful in clinical settings, as they lack the precise information ophthalmologists consider essential.

An ophthalmologist, when evaluating a digital fundoscopy of a diabetic patient at risk of developing diabetic retinopathy (DR), can be viewed as a classifier, analyzing regions of the image associated with pathological signs such as microhemorrhages, exudates, microaneurysms, and macular edema, among others. Once these elements are identified and their spatial relationships documented, the physician evaluates the data and produces a diagnosis, which is analogous to the output of a classification algorithm. However, the specific elements (weights) that determine the diagnosis are crucial in medical practice, as they directly influence the treatment plan of the patient and the survival prognosis.

While transformers can offer accurate predictions, they often fail to provide sufficient insight into how those predictions were reached, making clinical adoption difficult. This creates an opportunity for the development of intelligent computing models with explainable capabilities, where classifications can be traced back to the logical steps typically followed in clinical reasoning in everyday ophthalmological practice. Interpretability is a crucial aspect of machine learning, especially in sensitive applications like medicine. Investigating how to make models more understandable for humans allows researchers and professionals to better grasp the decision-making process. By focusing on interpretability, ethical development of artificial intelligence is promoted, ensuring that models are not only accurate but also transparent in their operation. Developing algorithms from this perspective would naturally facilitate collaborative, multidisciplinary, and integrated work between computational and medical experts, fostering the growth of computational medicine in clinical practice.

References

1. Wilkinson, C.P., Ferris III, F.L., Klein, R.E., Lee, P.P., Agardh, C.D., Davis, M., Dills, D., Kampik, A., Pararajasegaram, R., Verdaguer, J.T., *et al.:* Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology 110(9), 1677–1682 (2003)
2. Tenorio, G., Ramírez-Sánchez, V.: Retinopatía diabética; conceptos actuales **73**(3), 193–201
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. https://doi.org/10.48550/ARXIV.1706.03762. Publisher: [object Object] Version Number: 7. Accessed 2024-05-01
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv. Version Number:

2. https://doi.org/10.48550/ARXIV.2010.11929. https://arxiv.org/abs/2010.11929 Accessed 2024-09-17
5. Page, M.J., Moher, D., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., McKenzie, J.E.: PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews, 160. https://doi.org/10.1136/bmj.n160. Accessed 2024-09-18
6. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules, vol. 30, pp. 3856–3866. https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf
7. Yang, Y., Cai, Z., Qiu, S., Xu, P.: A novel transformer model with multiple instance learning for diabetic retinopathy classification **12**, 6768–6776. https://doi.org/10.1109/ACCESS.2024.3351473. Accessed 2024-09-10
8. Zhao, S., Wu, Y., Tong, M., Yao, Y., Qian, W., Qi, S.: CoT-XNet: contextual transformer with xception network for diabetic retinopathy grading **67**(24), 245003. https://doi.org/10.1088/1361-6560/ac9fa0. Accessed 2024-09-12
9. Sadeghzadeh, A., Junayed, M.S., Aydin, T., Islam, M.B.: Hybrid CNN+transformer for diabetic retinopathy recognition and grading. In: 2023 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–6. IEEE. https://doi.org/10.1109/ASYU58738.2023.10296789. https://ieeexplore.ieee.org/document/10296789/ Accessed 2024-09-12
10. Liu, Y., Yao, D., Ma, Y., Wang, H., Wang, J., Bai, X., Liu, Y., Zeng, G.: Stmf-Drnet: A Multi-Branch Fine-Grained Classification Model for Diabetic Retinopathy Using Swin-Transformerv2. https://doi.org/10.2139/ssrn.4930461. https://www.ssrn.com/abstract=4930461 Accessed 2024-09-12
11. Benabdessalam, N., Mabrouk, S., Ghorbel, F.: BERTImages for diabetic retinopathy classification. In: 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), pp. 1–7. IEEE. https://doi.org/10.1109/AICCSA59173.2023.10479275. https://ieeexplore.ieee.org/document/10479275/ Accessed 2024-09-11
12. Ainapur, S.S., Patil, V.: A novel automated system for early diabetic retinopathy detection and severity classification, 3593. https://doi.org/10.1002/cem.3593. Accessed 2024-09-09
13. Jagadesh, B.N., Karthik, M.G., Siri, D., Shareef, S.K.K., Mantena, S.V., Vatambeti, R.: Segmentation using the IC2t model and classification of diabetic retinopathy using the rock hyrax swarm-based coordination attention mechanism **11**, 124441–124458. https://doi.org/10.1109/ACCESS.2023.3330436. Accessed 2024-09-12
14. Huang, Y., Lyu, J., Cheng, P., Tam, R., Tang, X.: SSiT: Saliency-guided Self-supervised Image Transformer for Diabetic Retinopathy Grading. arXiv. Version Number: 5. https://doi.org/10.48550/ARXIV.2210.10969. https://arxiv.org/abs/2210.10969 Accessed 2024-09-12
15. Gu, Z., Li, Y., Wang, Z., Kan, J., Shu, J., Wang, Q.: Classification of diabetic retinopathy severity in fundus images using the vision transformer and residual attention **2023**(1), 1305583. https://doi.org/10.1155/2023/1305583. Accessed 2024-09-11
16. Khan, I.U., Raiaan, M.A.K., Fatema, K., Azam, S., Rashid, R.U., Mukta, S.H., Jonkman, M., De Boer, F.: A computer-aided diagnostic system to identify diabetic retinopathy, utilizing a modified compact convolutional transformer and low-resolution images to reduce computation time **11**(6), 1566. https://doi.org/10.3390/biomedicines11061566. Accessed 2024-09-09
17. Zang, F., Ma, H.: CRA-net: Transformer guided category-relation attention network for diabetic retinopathy grading **170**, 107993. https://doi.org/10.1016/j.compbiomed.2024.107993. Accessed 2024-09-12
18. Karthika, S., Durgadevi, M.: Improved ResNet_101 assisted attentional global transformer network for automated detection and classification of diabetic retinopathy disease **88**, 105674. https://doi.org/10.1016/j.bspc.2023.105674. Accessed 2024-09-12
19. Oulhadj, M., Riffi, J., Khodriss, C., Mahraz, A.M., Yahyaouy, A., Abdellaoui, M., Andaloussi, I.B., Tairi, H.: Diabetic retinopathy prediction based on vision transformer and modified capsule network **175**, 108523. https://doi.org/10.1016/j.compbiomed.2024.108523. Accessed 2024-09-12

20. Mathews, M.R., M., A.S.: Enhanced transformer model: A new frontier in diabetic retinopathy classification. In: 2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), pp. 334–338. IEEE. https://doi.org/10.1109/WIECON-ECE60392.2023.10456516. https://ieeexplore.ieee.org/document/10456516/ Accessed 2024-09-12
21. Zhang, Y., Ma, X., Huang, K., Li, M., Heng, P.-A.: Semantic-oriented visual prompt learning for diabetic retinopathy grading on fundus images **43**(8), 2960–2969. https://doi.org/10.1109/TMI.2024.3383827. Accessed 2024-09-12
22. Nazih, W., Aseeri, A.O., Atallah, O.Y., El-Sappagh, S.: Vision transformer model for predicting the severity of diabetic retinopathy in fundus photography-based retina images **11**, 117546–117561. https://doi.org/10.1109/ACCESS.2023.3326528. Accessed 2024-09-13
23. Wu, J., Hu, R., Xiao, Z., Chen, J., Liu, J.: Vision transformer-based recognition of diabetic retinopathy grade **48**(12), 7850–7863. https://doi.org/10.1002/mp.15312. Accessed 2024-09-13
24. Galappaththige, C.J., Kuruppu, G., Khan, M.H.: Generalizing to Unseen Domains in Diabetic Retinopathy Classification. arXiv. Version Number: 2. https://doi.org/10.48550/ARXIV.2310.17255. https://arxiv.org/abs/2310.17255 Accessed 2024-09-19
25. Kommaraju, R., Anbarasi, M.S.: Detection of diabetic retinopathy (DR) severity from fundus photographs using swin transformer. In: 2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), pp. 1–6. IEEE. https://doi.org/10.1109/ICRAIE59459.2023.10468500. https://ieeexplore.ieee.org/document/10468500/ Accessed 2024-09-12
26. Haque, M.M., Akter, S., Ashrafi, A.F.: SwinMedNet: Leveraging swin transformer for robust diabetic retinopathy classification from the RetinaMNIST2d dataset. In: 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), pp. 1286–1291. IEEE. https://doi.org/10.1109/ICEEICT62016.2024.10534544. https://ieeexplore.ieee.org/document/10534544/ Accessed 2024-09-12
27. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv. http://arxiv.org/abs/1505.04597 Accessed 2024-10-21
28. Takahashi, S., Sakaguchi, Y., Kouno, N., Takasawa, K., Ishizu, K., Akagi, Y., Aoyama, R., Teraya, N., Bolatkan, A., Shinkai, N., Machino, H., Kobayashi, K., Asada, K., Komatsu, M., Kaneko, S., Sugiyama, M., Hamamoto, R.: Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review **48**(1), 84. https://doi.org/10.1007/s10916-024-02105-8. Accessed 2024-10-21
29. Pantelaios, D., Theofilou, P.-A., Tzouveli, P., Kollias, S.: Hybrid CNN-ViT models for medical image classification. In: 2024 IEEE International Symposium on Biomedical Imaging (ISBI), pp. 1–4. IEEE. https://doi.org/10.1109/ISBI56570.2024.10635205. https://ieeexplore.ieee.org/document/10635205/ Accessed 2024-10-22

Segmentation of High Quality Histopathological Micrographs Using a Genetic Algorithm and Otsu's Method

F. J. Rojas-Pérez, E. G. Pérez-Pérez, Anabel Martínez-Vargas, and J. R. Conde-Sánchez

Abstract In medical areas, digital image processing allows for improved accuracy in detecting and diagnosing diseases, where better quality images contain large amounts of information that computationally make a difference in computer-aided diagnosis. Traditional methods from library functions in programming languages can be computationally expensive and slow to process. In the case of thresholding by Otsu's method applied to 8 bits per pixel (bpp) images, 256 calculations are made to determine the best threshold value. In the case of 16 bpp quality, images 65,536 different calculations are required, which increases operations number. This chapter presents a Genetic Algorithm (GA) based on Otsu's method to optimize the search for the best threshold value in image segmentation. A comparative analysis is performed using threshold value calculated by a standard function from *python opencv* library and threshold value calculated by GA, both applied to histopathological micrographs of 16 bpp breast cancer images. Results show that GA calculates an outstanding threshold value close to the optimal threshold calculated by the standard function with a negligible error rate of 0.376% (less than 1%) in the worst case for the image

These authors contributed equally to this work.

F. J. Rojas-Pérez · E. G. Pérez-Pérez
Fuentes de Energía y Sólidos Cristalinos Aplicados a Dispositivos Optoelectrónicos, Universidad Politécnica de Pachuca, Zempoala, Hidalgo, Mexico
e-mail: egonzalo@upp.edu.mx

A. Martínez-Vargas (✉)
Dirección de Investigación, Innovación y Posgrado, Universidad Politécnica de Pachuca, Zempoala, Hidalgo, Mexico
e-mail: anabel.martinez@upp.edu.mx

J. R. Conde-Sánchez
Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla, Ciudad Universitaria, Puebla, Mexico
e-mail: rconde@fcfm.buap.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_6

binarization process. We conclude that GA in conjunction with Otsu's method is a viable alternative for high-quality digital image processing, especially for images with higher bpp.

Keywords Otsu's thresholding · Genetic algorithm · 16-bpp images · Otsu's method

1 Introduction

Digital image processing (DIP) has evolved significantly since its inception and technological advances have impacted how we analyze high-quality digital images in medical areas, these images have vital health information, but they are increasingly complex. Consequently, there is a need for innovative approaches to solve problems that traditional methods can not address efficiently.

GA has emerged as an alternative tool in digital image processing tasks. These algorithms are based on search and optimization methodologies that allow finding optimal or approximate solutions to complex problems where possible solutions are difficult or impossible to compute. GAs are flexible and adaptable, which makes them a viable option for DIP, e.g., image segmentation [1], restoration [2], enhancement [3], etc.

In 1992, Dasgupta & McGregor [4] worked with a structured genetic algorithm (SGA) for automatic image registration, which is a process of superimposing two images of the same area or scene generated from different perspectives or at different periods. SGA differs in their ability to work with redundant genetic material and to implement activation mechanisms in the genes that make up the candidate solutions through multilayered structures in the chromosome representation. This allows changes in high-level genes, delimiting the search for large areas with high-potential solutions, and changes in low-level genes, which allows further exploration of the subarea. SGA has improved computational complexity compared to simple genetic algorithms, with better spatial scaling and robustness in the solution search process.

For 2006, Kim et al. [5] proposes an integrated face recognition system that uses attributes based on Gabor wavelets, which focus on spatial locality and orientation selectivity and are essential for face recognition. The problem with the Gabor wavelet method is the computational cost and the high sensitivity to illumination variations present in images. To overcome this problem, it was proposed that a fast genetic algorithm be used to optimize the parameters of orientations and Gabor scales by improving the feature selection and extraction process with better approximation to local image contour regions. The developed system efficiently selects the set of attributes for face detection and classification, by testing using the FERET dataset the system has a recognition rate of 97.27%.

In the year 2016, Khanna et al. [6] proposed using a genetic algorithm to determine the embedding strength and parameters associated with watermarking, attempting to create an optimal watermarking scheme for image copyright protection capable of

resisting attacks. In this regard, a GA was implemented to find the optimal value of the watermark scaling factor, which determines the size and visibility of the watermark influences the effectiveness of image protection, and limits or stops unauthorized use and distribution. Successfully, the GA selects acceptable values of watermark embedding strength with the balance between the parameters of peak signal-to-peak noise ratio and normalized correlation, which is fundamental in watermark development.

In 2018, Wang [7] proposed a method for infrared image segmentation that reduces morphological noise using an adaptive morphological weighting algorithm and enhanced by Otsu's method in conjunction with a genetic algorithm. Improved GA achieves balanced and diverse convergence in the population with accelerated performance due to the automatic adjustment of genetic control parameters according to the fitness and spread of the population. This methodology optimally calculates the threshold value for infrared image segmentation overcoming common GA problems such as poor convergence or premature convergence, among others.

In DIP, image binarization by thresholding [8] effectively differentiates the pixels corresponding to the region of interest (ROI) from the image background. Threshold selection [9], denoted as t, is crucial for correct binarization. Some techniques and methodologies allow calculating thresholds with acceptable binarization results in images and some cases determine the optimal threshold [10] for more effective image segmentation.

Otsu's method [11] is one of the most used and effective techniques for image binarization, this method calculates the optimal threshold by exhaustive search from frequency histogram in grayscale images, where, intra-class variance is the criterion to maximize that determine optimal threshold.

One disadvantage of Otsu's method [12] is that performs calculations for all possible gray values in a grayscale image, for example, a standard format in the grayscale image has a representation of 8 bpp, which means that can reproduce 2^8 or 256 different gray levels. With 8 bpp representation, the number of calculations is relatively small and fast to perform on most computer systems. However, in healthcare areas work with grayscale images with higher quality compared to the standard format, with representations of 2^{16} or $65,536$ different gray levels, considerably increasing the calculations and affecting the amount of time and computational resources required for thresholded image binarization using Otsu's method.

State-of-the-art contribution consists of providing an alternative methodology based on GA for high-quality image binarization. The motive of this methodology is to solve the problem of thresholding images with a high bpp representation, the traditional method consists of using standard functions whose calculations increase 2^n depending on the number of bpp. On the other hand, our approach reduces the computational complexity and calculations required for an effective image binarization process that easily adapts to the number of bpp, in this case, GA was adapted to high-quality 16 bpp images.

The remaining part of this chapter is composed as follows: Sect. 2 describes the problem of working with DIP and high-quality images, the proposed solution, and closes with the objective of this work. Section 3 presents the basic concepts to

understand image thresholding methodology. Section 4 discusses the methodology of simple image thresholding. Section 5 shows image thresholding by Otsu's method. Section 6 describes the development of a GA focusing on image thresholding using Otsu's method. Section 7 shows the results of comparing the threshold value obtained by GA as a whole of Otsu's method and the optimal threshold value obtained by Otsu's method from the Opencv library. Section 8 shows the results. Finally, Sect. 9 exposes the critical issues of the implementation of GAs in conjunction with digital image thresholding using Otsu's method.

2 Problem Statement

Otsu's method addresses optimization and search issues according to bpp used to represent high-quality images. Exhaustive threshold search on high-quality images requires more calculations and requires more hardware resources, which can be a limiting factor in systems with limited computational resources.

To treat this problem, metaheuristics have emerged as a method designed to solve problems faster and more competently than traditional methods, by compromising the optimization and accuracy of the final solution. Priority is given to a satisfactory solution in an acceptable time and fewer calculations. In this sense, GAs are taken into account as a viable tool for the optimization of algorithms in exhaustive search problems, such as Otsu's method.

On the other hand, GA uses a solution population and applies operators inspired by species evolution to explore the search space more efficiently, allowing optimal or near-optimal solutions to be found in a reasonable amount of time. GA can be adapted to the problem at hand as long as the objective function and constraints of the problem are defined. In turn, by omitting the need to evaluate every possible solution, GA reduces the computational costs associated with an exhaustive search.

3 Basic Concepts

A pixel is the smallest and most basic unit that makes up a digital image (see Fig. 1). Each pixel contains information about a homogeneous color and brightness intensity, by joining multiple pixels together using a matrix of pixels arranged in rows and columns, complex visual representations can be recreated. Complex images are defined when pixels are grouped in a 2-dimensional matrix arrangement, luminance intensity is represented in the grayscale model, which is a color model that represents images using only shades of gray, ranging from pure black to pure white, through a range of intermediate grays.

In computational issues, grayscale images (see Fig. 2) facilitate analysis and processing by allowing algorithms to operate quickly in image segmentation tasks.

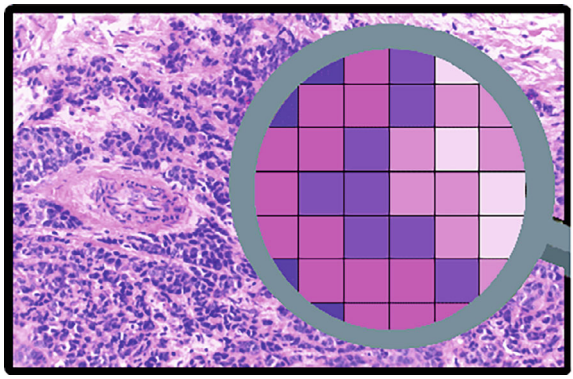


Fig. 1 Visual representation of pixels in a histopathologic micrograph

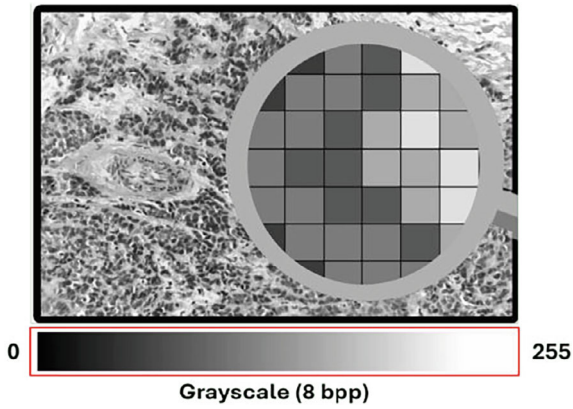


Fig. 2 Grayscale histopathologic micrograph

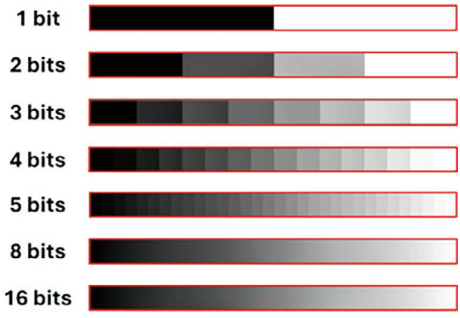


Fig. 3 Color accuracy by number of bpp

Bits per pixel (bpp) is the number of bits of information used to represent one color in each pixel in a digital image (see Fig. 3). The amount of bpp determines the number of different colors that can be displayed in an image. Computer systems are based on binary systems, therefore, color deep of n bits that can be represented 2^n possible colors, each bpp increases color accuracy and color gamut.

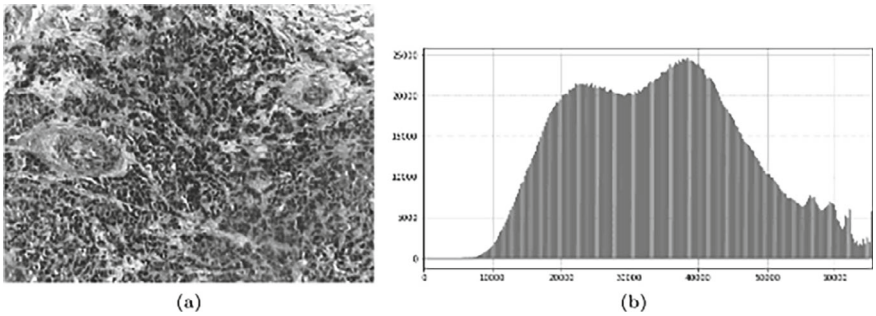


Fig. 4 **a** 16 bpp grayscale image, **b** frequency histogram of **(a)**

An image histogram is a graphical representation of the frequency and distribution of intensity levels (color) of pixels that make up the digital image (see Fig. 4). The horizontal axis represents pixel intensity values, ranging from pure black (0) to pure white (255) in 8 bpp images or pure black (0) to pure white (65535) in 16 bpp images. The vertical axis indicates the frequency of pixels in each intensity level.

4 Image Thresholding

It is a digital image process that converts color or grayscale images to binary images by grouping pixels (see Fig. 5), which facilitates the identification of regions and objects of interest. Thresholding consists of defining a gray level value as a threshold to separate pixels of an image into different classes. These classes are modified from their original intensity level to values of zero and one and classified into two categories: object or region of interest (foreground) and background.

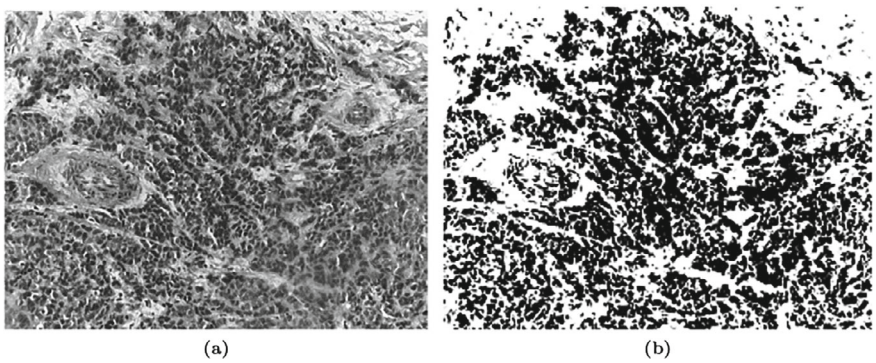


Fig. 5 **a** Grayscale image, **b** Image binarization by thresholding

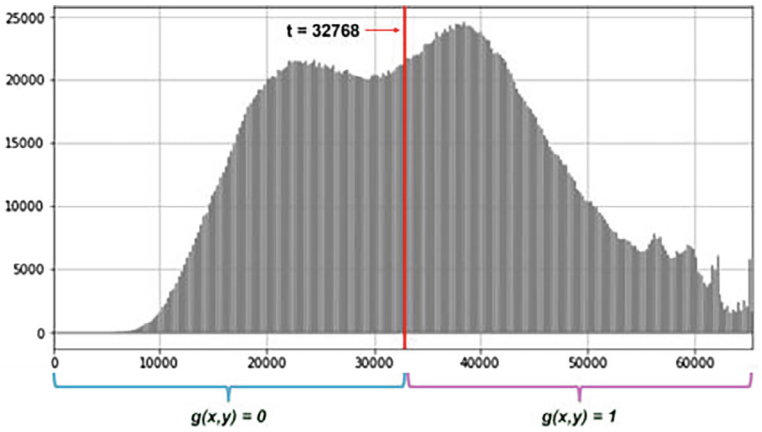


Fig. 6 Visualization of image binarization by thresholding

The thresholding process can be expressed mathematically. Let $f(x, y)$ be a 2D digital image, (x, y) represents spatial coordinates and $f(x, y)$ represents color intensity level. A threshold value is defined to group pixels of any image in two classes, this threshold is denoted as t. Then, a binary image $g(x, y)$ is defined as:

$$g(x, y) = \begin{cases} 0 & f(x, y) \leq t \\ 1 & f(x, y) > t \end{cases} \tag{1}$$

where $g(x, y) = 0$ and $g(x, y) = 1$ are the classes of pixels representing the region of interest and pixels corresponding to the background of the image, as shown in Fig. 6.

5 Image Segmentation by Otsu's Method

Threshold t selection is crucial for good segmentation. Automatic methods such as Otsu seek optimal thresholds according to intensity levels and based on statistical criteria. Otsu's method seeks a threshold t that maximizes variance between classes corresponding to the object of interest and background.

Otsu's method is defined mathematically as follows. Let L be the total number of gray levels in the image (65536 gray levels for 16 bpp images). i represents a gray level such that $i \in L | 0 \leq i \leq L - 1$ and n_i is the number of pixels with intensity level i.

Otsu's method is an iterative process that varies the threshold t value, starting with $t = 0$ until $t = L - 1$, which divides pixels into two classes denoted as $C_1(t)$ y $C_2(t)$, and consequently changes variances value each time threshold t changes.

The intensity probability of gray level i is calculated as:

$$p_i = \frac{n_i}{N}, \; p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \tag{2}$$

where N is the total number of pixels in the image.

Class averages $C_1(t)$ and $C_2(t)$ are defined as:

$$\mu_1(t) = \sum_{i=0}^{t-1} \frac{ip_i}{w_1(t)} \tag{3}$$

$$\mu_2(t) = \sum_{i=t}^{L-1} \frac{ip_i}{w_2(t)} \tag{4}$$

where $w_1(t)$ y $w_2(t)$ are cumulative probabilities of $C_1(t)$ and $C_2(t)$ classes:

$$w_1(t) = \sum_{i=0}^{t-1} p_i \tag{5}$$

$$w_2(t) = \sum_{i=t}^{L-1} p_i \tag{6}$$

The global mean of the image is calculated as:

$$\mu_T = w_1(t)\mu_1(t) + w_2(t)\mu_2(t) = \sum_{i=t}^{L-1} ip_i \tag{7}$$

Variance between $C_1(t)$ and $C_2(t)$ is established as:

$$\sigma_b^2(t) = w_1(t)(\mu_1(t) - \mu_T(t))^2 + w_2(t)(\mu_2(t) - \mu_T(t))^2 \tag{8}$$

Then, Variance within $C_1(t)$ and $C_2(t)$ classes is defined as:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \tag{9}$$

$\sigma_1^2(t)$ and $\sigma_2^2(t)$ are variances of $C_1(t)$ and $C_2(t)$ classes:

$$\sigma_1^2(t) = \sum_{i=0}^{t-1} \frac{(\mu_T - \mu_1(t))^2 p_i}{w_1(t)} \tag{10}$$

$$\sigma_2^2 = \sum_{i=t}^{L-1} \frac{(\mu_T - \mu_2(t))^2 p_i}{w_2(t)} \tag{11}$$

Otsu's method selects the optimal threshold according to $\sigma_w^2(t)$ and $\sigma_b^2(t)$ values that provide the highest ratio of inter-class variance to intra-class variance. Thus, Eq. 12 defined as the quotient of between-class variance and within-class variance is used as an argument that determines the most appropriate threshold t.

$$Q(t) = \frac{\sigma_b^2}{\sigma_w^2} \tag{12}$$

By maximizing $Q(t)$, a threshold is ensured that effectively separates the classes while minimizing variability within each class. Therefore, Otsu's method seeks to maximize $Q(t)$, which provides the largest statistical separation between two classes as a criterion for optimal threshold selection:

$$t^* = argmax_t Q(t) \tag{13}$$

6 Genetic Algorithm

One disadvantage of Otsu's method is that it is an exhaustive search method. This means that although this method ensures that an optimal threshold is found, it is inefficient, and has a high computational cost for high-quality images (e.g. 16 bpp images). In this regard, GA is an approach for problem optimization that finds effective solutions in a short period, often, at the expense of optimal.

A GA [13] is a metaheuristic that simulates biological evolution to find solutions for optimization problems. Scanning t threshold suitable for image binarization can be formulated as a search and optimization problem [14], where, candidate solutions are expressed in binary arrays and each iteration modifies candidate solutions similar to the evolutionary process and natural selection of individuals in a population. Algorithm 1 shows the flowchart of a basic GA [15], where, the performance of GA is based on three important points:

- Independent sampling provided by randomly initialized large populations.
- Selection of the most suitable individuals by delimiting the search area to the greatest suitability regions.
- Partial solutions crossing to explore search space and accelerate the optimization process.

Algorithm 1: Basic GA

```
Initialize population
while unemployment condition is not fulfilled do
    Fitness evaluating of each individual
    Parent selection
    Crossing
    Mutation
    Integrate new individuals into the population
Selection of best individual
```

6.1 Population Initialization

The basic GA version [16], initializes a population with random individuals, these individuals are composed of a 16-bit binary array which represents a candidate solution for searching problem of optimal threshold for high-quality image segmentation (see Fig. 7). Each bit array element that represents one individual is called a gene and has only two possible values 0 and 1 (see Fig. 8).

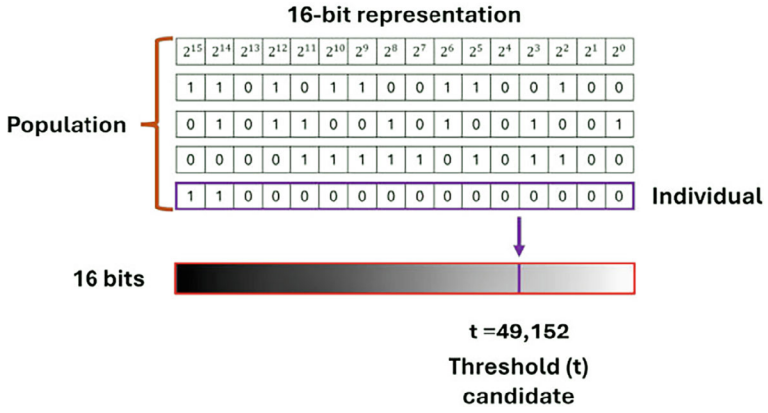


Fig. 7 Representation of individuals for the GA

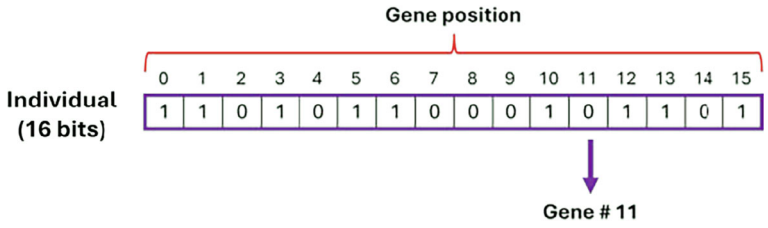


Fig. 8 Visualization of a gene in an individual for the GA

Genetic differences between individuals in the population make some candidate solutions more viable than others, in other words, there are superior and inferior individuals and they represent better or worse solutions to optimization problems.

At the programming level, Algorithm 2 describes how to initialize the 16-bit random population of individuals.

Algorithm 2: Initialization of random population

Input: N
Output: P
for *Individual in range* $(0, N-1)$ **do**
  Generate a random integer in the range (0, 65535)
  Generate binary representation of random number
  Assigning binary representation to a new individual
  Integrate new individual into population

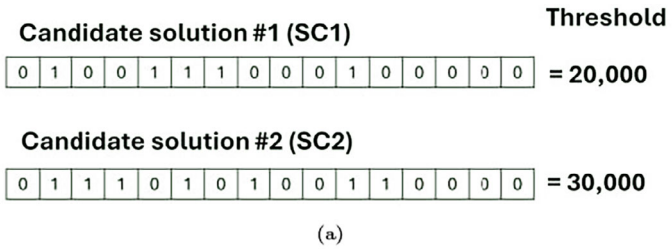
where:

- N is the number of individuals that will make up the population.
- P is the initialized population with random individuals.

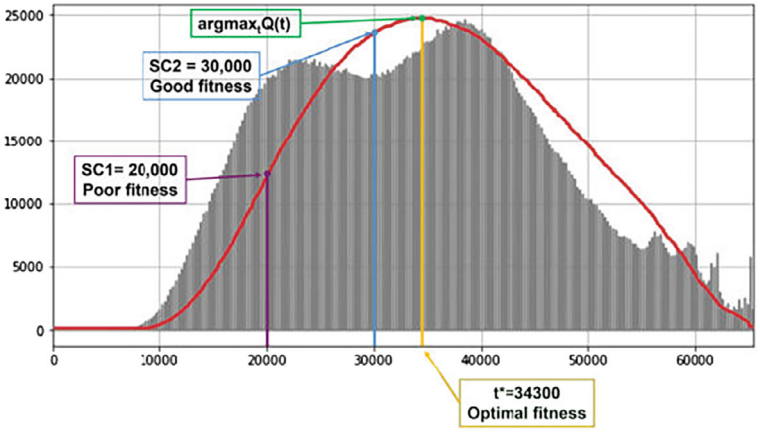
6.2 Fitness Measurement

GA requires an objective function, which is a mathematical expression whose value is to be maximized or minimized and is formulated according to the problem posed. For the exhaustive search problem of Otsu's method using a GA, $Q(t)$ in Eq. 12 is the argument to maximize to find optimal threshold t^* in Eq. 13. $Q(t)$ defines the objective function and determines how good or bad the candidate solution is.

Figure 9 shows two individuals with a 16-bit representation focused on a threshold optimization problem, both candidate solutions evaluated with objective function in Eq. 12.



(a)



(b)

Fig. 9 Example of fitness evaluation of two individuals for a GA **a** Representation of individuals, **b** Visualization using a histogram of candidate solutions

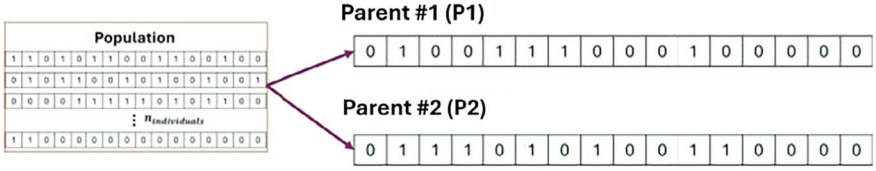


Fig. 10 Random selection of parents for a GA

6.3 Parent Selection

This operator randomly selects a pair of individuals from a previously initialized population and subsequently carries out a crossing operation between individuals (see Fig. 10). The better the fitness of an individual, the more likely that fitness will be selected for crossing, and endure in future generations.

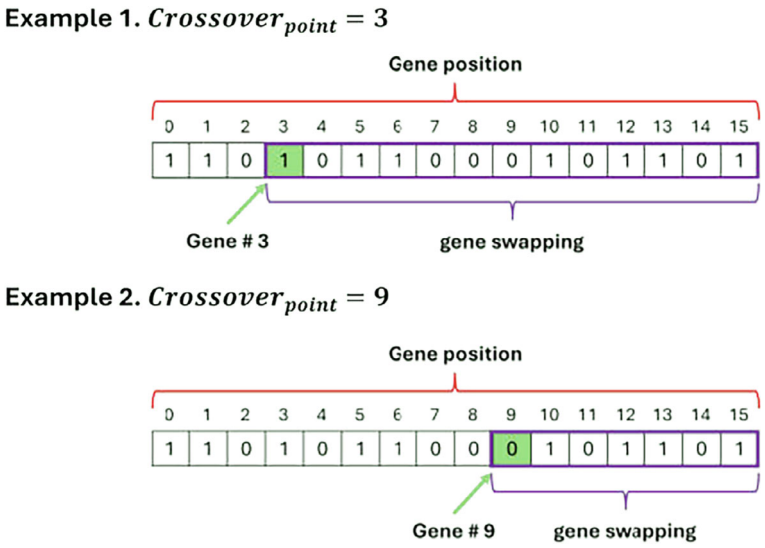

Example 1. $Crossover_{point} = 3$
Gene position
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 1 0 1 0 1 1 0 0 0 1 0 1 1 0 1
Gene # 3
gene swapping
Example 2. $Crossover_{point} = 9$
Gene position
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 1 0 1 0 1 1 0 0 0 1 0 1 1 0 1
Gene # 9
gene swapping

Fig. 11 Examples of crossover point selection for a GA

A better approach is tournament selection [17], which creates a randomly selected subset of individuals smaller than the population size where individuals will be compared with each other to perform the crossover and mutation operators to determine the most suitable individuals.

Tournament selection allows a balanced exploration in solution space with an elite approach where the best individuals endure in future generations.

6.4 Crossing

Crossing operation is performed according to a percentage of crossover probability (commonly 80%) to previously selected individuals and performs gene swapping to create two new individuals called children. Different methods perform crossover between individuals, each method explores different combinations of features and optimizes the search for effective solutions.

Crossing by a point [17] is the basic crossing method to generate children from two previously selected parents. A random integer is determined in a range in which an individual's length. For a 16-bit individual, points to perform crossing must be in the range [0, 15] (see Fig. 11). Determining the crossover point will allow the exchange of genes between parents.

After crossover point selection, proceed with gene swapping. Let P_1 and P_2 be the parents and $C_P = 5$ the crossover point as Fig. 12 shows. Child 1 as C_1 inherits the

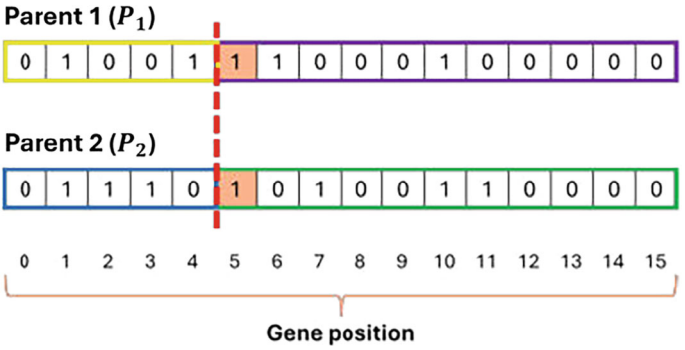


Fig. 12 P_1 and P_2 parents for crossing operation

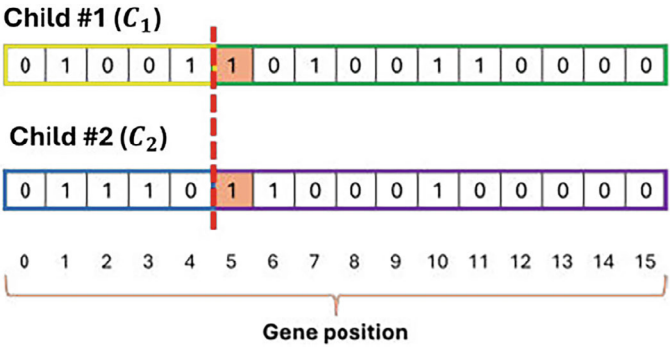


Fig. 13 Gene inheritance for children C_1 and C_2 by P_1 and P_2 parent

first segment of P_1 and the second segment of P_2, meanwhile, child 2 as C_2 inherits the first segment of P_2 and the second segment of P_1 (see Fig. 13).

6.5 Mutation

After the crossover operation, a mutation process is performed on children to introduce additional variability and explore new areas of search space. Commonly a probability between 0.01 and 0.1, makes modifications to an individual's genes. The basic mutation process determines mutation probability for each gene of child and in case of mutation, a specific operator is applied according to the representation of individuals. For binary representations, the bit-flip method [17] is characterized by its simplicity, involving a change in gene value from 0 to 1 and vice versa (see Fig. 14).

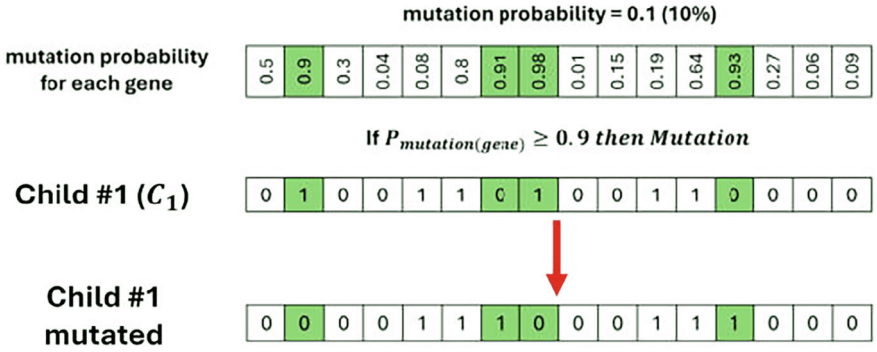

mutation probability = 0.1 (10%)
mutation probability for each gene
0.5 0.9 0.3 0.04 0.08 0.8 0.91 0.98 0.01 0.15 0.19 0.64 0.93 0.27 0.06 0.09
If $P_{mutation(gene)} \geq 0.9$ then Mutation
Child #1 (C_1)
0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 0
Child #1 mutated
0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0

Fig. 14 Representation of gene mutation process

6.6 *Integration Process*

New individuals (C_1 and C_2) generated by crossover and mutation operations have a high probability of being good solutions. A generation comprises stages of parent selection, genetic operations, and fitness selection in the population; with each generation, the general fitness of the population improves. Different approaches simulate how children replace parents. The simplest approach is for offspring to replace parents in the direct line of descent throughout the population (see Fig. 15). While this type of total replacement is effective in large populations, it consequently results in significant losses of the best candidate solutions.

Instead of total replacement, a partial replacement can be made by an elitist selection between parents and children (see Fig. 16) to conserve the best individuals. This elitist approach keeps individuals with better fitness, which allows elite individuals to last in future generations, ensuring that the best fitness is not missed.

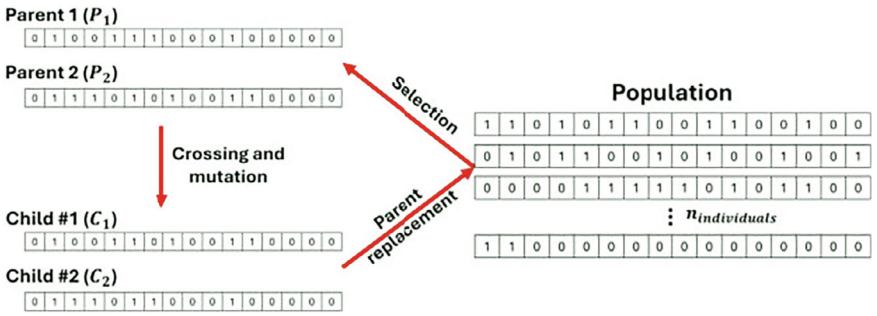


Fig. 15 Direct replacement of individuals in the population

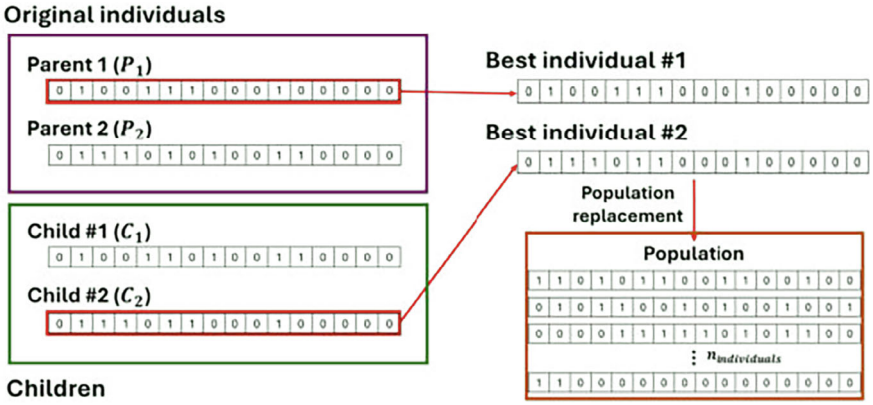


Fig. 16 Elitist replacement of individuals in the population

7 Results

To evaluate GA efficiency to find the optimal or near-optimal threshold, a comparative analysis is performed using *cv2.threshold* function from *OpenCv Python* library, which, calculates the optimal threshold of an image using Otsu's method.

A meaningful comparison between the standard Python library function and GA involves calculating the threshold for numerous images by both resources and analyzing them to obtain reliable results. Therefore, 100 images from the image bank *ICIAR 2018 BACH Challenge* were analyzed, which, is a public dataset used to evaluate breast cancer diagnostic algorithms. These images are in *tiff* format with 48 bpp color depth, that is to say, 16 bpp each *RGB* channel.

Each image is preprocessed before comparative analysis, this stage consists of separating the green channel (G) to be used in optimal threshold calculation (see Fig. 17).

Table 1 shows an analysis of 17 images with the greatest intensity levels difference, defined by threshold value calculated by GA (approximate threshold) and *OpenCv* library (optimal threshold).

The worst case out of 100 images analyzed is that of image *n021.tiff*, which has a difference of 247 gray levels from the optimum threshold. To determine whether this case has a significant error when working with 65536 different gray levels in a 16 bpp image, it is important to consider several factors:

- **Total Gray Levels**: In 16 bpp images there are 65,536 possible gray levels (from 0 to 65,535) so that each intensity level of difference with optimal threshold represents an error of $\frac{1}{65536} \approx 0.00000152$ approximately.

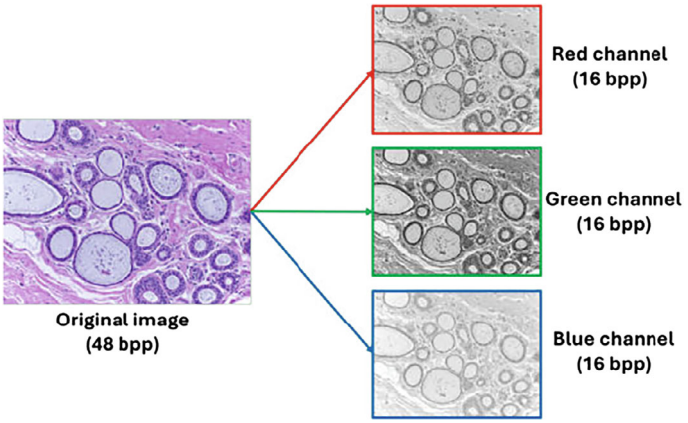


Fig. 17 R, G, and B channel display of a color image

Table 1 Comparative analysis of the worst approximate solutions

Image ID	Threshold by GA	Optimal threshold by library	Intensity difference	Error rate (0–100 %)
b017.tiff	46133	46118	15	0.0229
b023.tiff	45371	45347	24	0.0366
b025.tiff	40203	40242	39	0.0595
is004.tiff	42537	42520	17	0.0259
is008.tiff	41520	41492	28	0.0427
is020.tiff	43312	43287	25	0.0381
is021.tiff	41825	41795	30	0.0458
iv001.tiff	39454	39436	18	0.0275
iv010.tiff	42621	42571	50	0.0763
iv020.tiff	41506	41477	29	0.0443
n001.tiff	50509	50473	36	0.0549
n008.tiff	50197	50186	11	0.0168
n009.tiff	41261	41235	26	0.0397
n020.tiff	48199	48174	25	0.0381
n021.tiff	49151	49398	247	0.3769
n024.tiff	42274	42259	15	0.0229
n025.tiff	46901	46889	12	0.0183

- **Error in intensity levels**: *n021.tif* image is the most severe case with an error of 247 gray levels of difference from the optimum value. Error percentage is calculated as follows:

$$\text{Error percentage} = \frac{247}{65536} * 100 \approx 0.376\% \tag{14}$$

The percentage of error calculated in the worst case can have an impact, or not, depending on the level of accuracy required, for example:

- **High precision application**: A 247 gray level error is not considered a good approximation since the difference in levels is significant compared to total possible values. This error percentage may be noticeable and affect the results of image processing. However, if extreme precision is not required, solution approximation can be considered acceptable.
- **Low precision application**: A 247 gray level error is considered acceptable. A percentage error of 0.376% is not perceptible or does not affect the visual quality of the final result in image processing. When compared to the total range of 65,536 intensity levels, 247 levels of error is a negligible percentage, which makes the approximate solution good enough.

8 Discussion

Being image *n021.tiff* worst case with a difference of 247 gray levels from optimal value with an error of approximately 0.376%, it can be concluded that the error is within the allowable range for the image thresholding process. Figure 18b shows *n021.tiff* with optimal threshold, in contrast, Fig. 18c shows the image with the threshold found by the GA. Both thresholds capture regions of interest with high quality, so visually there is no difference between Fig. 18c and c. GA threshold approximation has been verified, which ensures that methodology based on a GA with Otsu's method works efficiently for high-quality image processing.

Given that the worst case of 100 processed images has demonstrated a relatively small error and is within permissible, it is important to note that any other case is significantly distant from being regarded as a failure. This indicates that the GA-based methodology and Otsu's method are robust and reliable since the error observed in the worst case is lower. In addition, consistency of results in other cases supports the idea that the methodology proposed offers a high degree of certainty in the calculation of appropriate or optimal threshold with an extremely low risk in case of failure in normal situations.

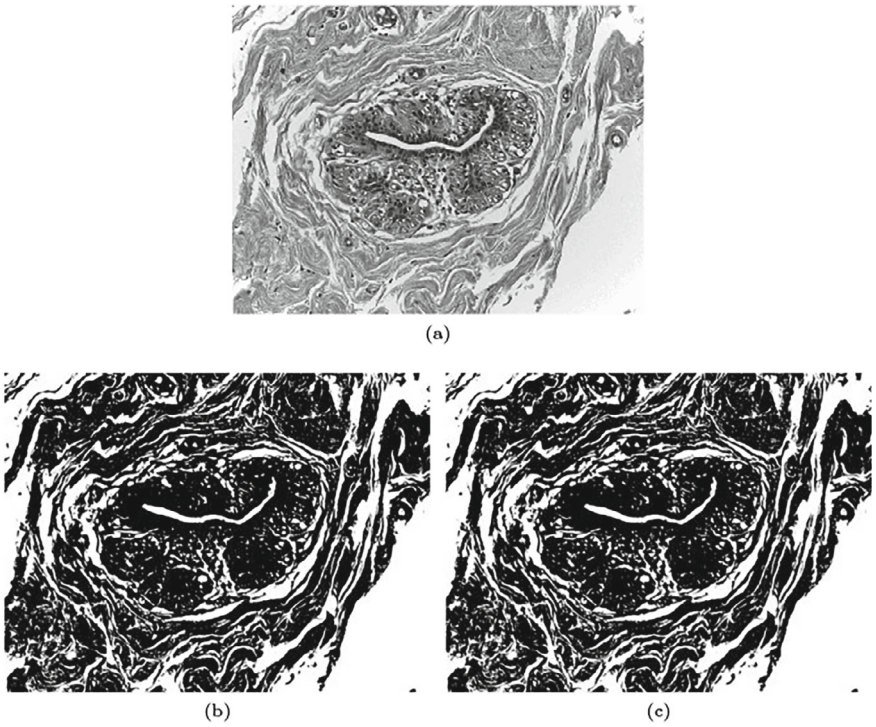


Fig. 18 16 bpp image thresholding **a** Original image (green channel), **b** Optimal threshold calculated by *OpenCv* library ($t = 49398$), **c** Threshold calculated by a GA ($t = 49151$)

9 Conclusions and Future Work

The higher the bpp, the greater the number of computations performed by the standard libraries while the GA with fewer calculations determines the appropriate threshold saving computer resources and execution time. GA in conjunction with Otsu's method can determine suitable or optimal threshold values for high-quality image binarization with an error level of 0.376% (less than 1%) in the worst case analyzed.

These results validate the performance of GA while opening new opportunities for its application in medical image processing, where evolutionary techniques help to identify and classify tumor lesions for early and effective diagnosis and treatment.

Future work includes improving the GA methodology with Otsu's method for adaptive thresholding of high-quality 16-bpp breast cancer histopathological micrographs. Adaptive thresholding through a GA will allow the segmentation of images with complex visual content with efficiency in the face of illumination variations.

There are also plans to integrate quantum computing into the development of GA due to its potential to enhance efficiency and effectiveness in addressing complex

problems. For instance, quantum computing enables unstructured searches, allowing the identification of optimal solutions within significantly larger search spaces. This capability accelerates the convergence towards optimal solutions, representing a substantial advancement in the fields of machine learning and artificial intelligence.

Declarations

The authors declare that there is no conflict of interest between us in relation to this research.

Acknowledgements F. J. Rojas-Pérez received a grant from the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT), CVU: 1078268.

References

1. Geng, J., Jia, D., Li, X., Zhang, C., Wu, N., Li, Z., He, Z.: Immunohistochemical image segmentation with improved aga-otsu. In: Sixteenth International Conference on Digital Image Processing (ICDIP 2024), vol. 13274, pp. 513–522 (2024). SPIE
2. Wang, Z., Pen, H., Yang, T., Wang, Q.: Structure-priority image restoration through genetic algorithm optimization. IEEE Access 8, 90698–90708 (2020)
3. Hashemi, S., Kiani, S., Noroozi, N., Moghaddam, M.E.: An image contrast enhancement method based on genetic algorithm. Pattern Recognition Letters 31(13), 1816–1824 (2010)
4. Dasgupta, D., McGregor, D.R.: Digital image registration using structured genetic algorithm. In: Neural and Stochastic Methods in Image and Signal Processing, vol. 1766, pp. 226–234 (1992). SPIE
5. Kim, D.-S., Jeon, I.-J., Lee, S.-Y., Rhee, P.-K., Chung, D.-J.: Embedded face recognition based on fast genetic algorithm for intelligent digital photography. IEEE Transactions on Consumer Electronics 52(3), 726–734 (2006)
6. Khanna, A.K., Roy, N.R., Verma, B.: Digital image watermarking and its optimization using genetic algorithm. In: 2016 International Conference on Computing, Communication and Automation (ICCCA), pp. 1140–1144 (2016). IEEE
7. Wang, Y.: Improved otsu and adaptive genetic algorithm for infrared image segmentation. In: 2018 Chinese Control And Decision Conference (CCDC), pp. 5644–5648 (2018). IEEE
8. Yousefi, J.: Image binarization using otsu thresholding algorithm. Ontario, Canada: University of Guelph **10** (2011)
9. Liu, D., Yu, J.: Otsu method and k-means. In: 2009 Ninth International Conference on Hybrid Intelligent Systems, vol. 1, pp. 344–349 (2009). IEEE
10. Goh, T.Y., Basah, S.N., Yazid, H., Safar, M.J.A., Saad, F.S.A.: Performance analysis of image thresholding: Otsu technique. Measurement 114, 298–307 (2018)
11. Bangare, S.L., Dubal, A., Bangare, P.S., Patil, S.: Reviewing otsu's method for image thresholding. International Journal of Applied Engineering Research 10(9), 21777–21783 (2015)
12. Bindu, C.H.: An improved medical image segmentation algorithm using otsu method. International Journal of Recent Trends in Engineering 2(3), 88 (2009)
13. Forrest, S.: Genetic algorithms. ACM computing surveys (CSUR) 28(1), 77–80 (1996)
14. Mitchell, M.: Genetic algorithms: An overview. In: Complex., vol. 1, pp. 31–39 (1995). Citeseer

15. Kramer, O.: Genetic Algorithms. Springer, Berlin (2017)
16. Lambora, A., Gupta, K., Chopra, K.: Genetic algorithm-a literature review. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMIT-Con), pp. 380–384 (2019). IEEE
17. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing, 2nd edn. Springer, Berlin (2015)

SDG 9 Industry, Innovation, Technology and Infrastructure

AI-Based Algorithm for the Automatic Detection of Missing Solder Balls in Encapsulated Semiconductors

Nestor A. Zermeño Campos, Julio C. Rodriguez Quiñonez, and Carlos A. Aguilar Avelar

Abstract The manufacturing of semiconductors is becoming increasingly expensive and complex, while market demand continues to grow. This makes the automation and optimization of the production process more crucial. The inspection stage is a critical point that is susceptible to improvement, as it still relies on visual inspection by specialized technicians. This results in increased production costs and time, as well as a higher risk of errors due to fatigue and bias affecting the performance of inspection personnel. In this context, the design and implementation of a new automatic inspection algorithm for detecting missing solder balls in encapsulated semiconductors is presented. This algorithm utilizes computer vision and deep learning. The images used for the training and testing of the models come from semi-automated inspection equipment that is part of the production process, and the obtained results demonstrate the feasibility of integrating such algorithms in a realistic production environment.

Keywords Semiconductor · Solder ball · Solder sphere · Ball detection · Missing ball · cnn · Mobilenet · vgg16

1 Introduction

The semiconductor industry is an ever-growing and highly competitive sector. Total semiconductor sales are expected to grow at a compound annual growth rate of 7.1% between 2021 and 2026, following strong growth of 11.0% between 2016 and 2021 [1]. In addition, the semiconductor market is projected to reach an impressive

N. A. Zermeño Campos · J. C. Rodriguez Quiñonez · C. A. Aguilar Avelar (✉)
Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Mexico
e-mail: aguilarav@uabc.edu.mx

N. A. Zermeño Campos
e-mail: alexander.zermeno@uabc.edu.mx

J. C. Rodriguez Quiñonez
e-mail: julio.rodriguez81@uabc.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_7

size of USD 168.3 billion by 2027, driven by the growing adoption of IoT devices and the increase in the number of data centers [2], as well as the demand for advanced technologies such as artificial intelligence and 5G [3].

Therefore, the semiconductor manufacturing industry presents significant challenges, especially due to the increasing volume of production, the small size of the components, and the precision required in their operation. An essential component of this process is the correct application of solder spheres in the fabrication of encapsulated semiconductors, which ensure the electrical connection between the semiconductor and the printed circuit board. The absence of these spheres can lead to device malfunction, which result in significant economic losses [4].

Conventional visual inspection of semiconductors, which relies on a trained human operator, exhibits a defect identification accuracy between 80% and 90%. However, after the first 30 min of work, the visual acuity of the human operator decreases significantly, especially for the analysis of a single defect type [5]. These methods can be slow and prone to human error. In addition, it is expected that the accuracy of manual or semi-automatic defect identification methods will decrease due to the continuous miniaturization of semiconductors, while the occurrence of soldering defects is expected to increase, since it can be caused by poor planning, incorrect electrode selection, manufacturing speed, equipment calibration, or current intensity [6].

Therefore, the inspection stage is a critical and improvement-prone point that still relies heavily on visual inspection of parts by specialized technicians. This method results in increased cost and production time, in addition to being prone to errors due to factors such as fatigue and bias that affect the performance of inspection personnel [5].

This work introduces the use of two pretrained VGG16 and MobileNet models for the automated detection of missing solder balls, in addition to a new custom model that is particularly designed to perform this task. Additionally, an innovative technique is introduced that employs RGB channels to store different enhanced features of the original image, improving classification performance. The evaluation of the presented models includes variations in hyperparameters such as epochs, batch size, learning rate and image size, with the aim of optimize model performance.

The structure of this chapter is organized as follows: Sect. 2 presents a selection of related literature. Section 3 introduces the main methods developed. Section 4 describes the performed experiments and obtained results. Section 5 discusses the obtained results. Finally, Sect. 6 presents the conclusions and future work.

2 Related Literature

There are commercially available automated inspection systems that use a combination of computer vision and machine learning techniques to detect and classify defects in semiconductor chips at various stages of manufacturing. Some of the major manufacturers of these systems include Applied Materials Inc, ASML Holding N.V., C&D

Semiconductor Services Inc, Hitachi High-Technologies Corp, KLA Corporation, Lasertec Corporation, Nikon Metrology NV, Onto Innovation, Inc, Thermo Fisher Scientific Inc, Toray Engineering [7]. Although traditional computer vision methods have had some success in detecting these defects, there is still room for improvement from specific challenges due to variability in defect location and shape [8].

Defect detection in semiconductors is a critical field due to the need to ensure the quality and reliability of electronic devices. Defects can be introduced at various stages of the manufacturing process, from circuit formation to final assembly. Traditional visual inspection methods and automated optical inspection systems have been useful, but face limitations in resolution and the ability to detect small and complex defects. For example, inspection of semiconductor wafers to detect surface defects combining traditional computer vision techniques and a lightweight CNN achieving 99.3% [9].

In addition, there are studies that seek to explore and solve specific cases presented by industry. A recent study introduced an improved algorithm, PCB-DeepLabV3, for segmenting vacuum defects in X-ray images of chip solder joints. This method, using MobileNetV2 and AMTPNet, significantly outperforms traditional visual inspection and classical segmentation models, achieving an average mIOU accuracy of 81.69% and a frame rate of 78.81 FPS [10]. However, this approach requires X-ray technology for imaging, which is expensive and hazardous for human technicians.

The detection of defects in solder spheres is also an area of considerable interest. Defects in solder joints can compromise the structural and functional integrity of electronic devices. Non-destructive inspection (NDT) techniques, such as radiography and ultrasonography, are indispensable tools for detecting internal defects without compromising solder integrity. These techniques allow the identification of internal and external discontinuities, such as slag inclusions and porosity, which can affect the quality of the weld.

In [6], Schlosser et al. (2022) introduced a hybrid multistage system of stacked deep neural networks (SH-DNN) for automated visual fault inspection in semiconductor manufacturing. This system combines classical image processing approaches with deep learning techniques, enabling the detection of fine structures within high-resolution images. The SH-DNN system focuses processing on regions of interest, allowing subsequent deep neural networks to classify defects with higher accuracy. The evaluation showed that the SH-DNN-based system significantly outperforms conventional and single DNN-based approaches, achieving an F1-score improvement from 95% to 98.9% and from 95.7% to 99.5%, respectively. The system also demonstrated real-time processing capabilities with runtimes of only 5.2 ms per chip image sample on consumer-grade hardware.

In [11], Chien et al. (2020) conducted experiments on 256 × 256 pixel wafer images, classifying them into four classes with known causes and comparing them with classic methods, such as SVM, logistic regression and random forest. The proposed CNN architecture, with 19,112 training samples, outperformed these methods in precision, accuracy, recall, F-Measure and AUC, achieving an average accuracy of 98.88%. In addition, transfer learning with pretrained faster-R-CNN models was explored, showing that this technology is viable for defect classification on wafers.

Although the CNN system proved to be an effective alternative to manual inspection, areas for improvement were identified, such as the inclusion of more defect types and a larger number of training samples.

In [12], Zhang et al. (2022) proposed four models for solder joint defect detection in X-ray images of industrial printed circuit boards. They used two pre-processing methods to address problems of slice number variation and incorrect region of interest (ROI). Two of the models were based on CNN, which are best for low Recall requirements, while the other two were based on LSTM, which is best for high Recall. All four models showed similar performance in terms of AUROC. The author mentions that requirements such as Recall ≥ 0.90 and Recall ≥ 0.95 are expected in the industry and that the introduction of AI in production can significantly reduce the workload of specialists. The fusion of models is suggested to improve the overall performance.

In [13], López de la Rosa et al. (2021) conducted a comprehensive review on the detection and classification of semiconductor defects from scanning electron microscope (SEM) images using machine learning (ML) and deep learning (DL) approaches. The review ranged from SEM fundamentals to typical ML and DL methods, classifying them into supervised, unsupervised and semi-supervised. The components of a CNN and the most common models were presented, as well as single and two-stage defect detection approaches. The review concluded that, although only nine relevant articles were found, there is a great opportunity to improve the results obtained so far through exploitable approaches. The information gathered will help future work to discard low-performing methods and focus on higher-performing ones, guiding the authors towards the most effective methods to improve defect detection and classification in the semiconductor industry .

However, most of the existing research focuses on detecting defects on wafer surfaces using advanced microscopy techniques, such as electron microscopy or X-ray imaging. While these methods are highly effective, they are also expensive and complex to implement in real production environments [14, 15]. In contrast, this work targets the detection of specific defects in solder ball grid arrays by employing deep learning models trained on grayscale images. These images can be captured using a standard computer vision-enabled camera, making our system practical, cost-effective and feasible for implementation in production lines.

3 Methods

This section presents a detailed description of the proposed models, the datasets used for their training, and the procedure for training an testing.

3.1 Raw Dataset

This study used 51 photographs of complete solder ball arrays, each of which had at least one case of a missing solder ball. These images were provided by an internationally renowned semiconductor company. To protect the company's privacy and intellectual property, the images were carefully edited to hide logos, serial numbers, labels and any other identifying details.

The images captured are the result of the reflection of light emitted by the materials when exposed to a controlled light source. This method makes it possible to highlight the differences between metallic and non-metallic materials present in the weld spheres. For this reason, the photographs were taken in grayscale, which facilitates the visual identification of the different types of materials and enhances the contrast between them.

The use of monochromatic imaging is particularly effective in semiconductor inspection, as it eliminates distractions caused by colour and focuses on variations in brightness and texture that are indicative of defects. In addition, this allows better detection of irregularities on the surface of solder spheres, such as the absence of a sphere, which could compromise the integrity of the device [16, 17].

To ensure accuracy and consistency in image capture, advanced illumination and photography techniques were employed. These techniques include the use of diffused light sources to minimise reflections and shadows, as well as camera calibration to ensure that all images were taken under uniform lighting conditions to obtain reliable and comparable data for subsequent analysis [18].

3.2 Solder Ball Segmentation

In this stage, we seek to identify the appropriate threshold level to reduce or eliminate the visibility of non-metallic materials in monochromatic images, which are the result of the light reflection emitted by the materials of the part to be analyzed. This threshold is determined by an exploratory analysis of each of the images of the set of parts.

The objective is to find a threshold value that allows to effectively separate metallic from non-metallic materials. Initially, this threshold was determined manually by observing the distribution of pixel intensities in the images and selecting a value that maximizes the separation between the pixels corresponding to metals and non-metals through image binarization. This threshold value is expected to remain constant as images are acquired under controlled lighting conditions and fixed camera settings.

3.2.1 Binarization

Binarization is a process that converts a grayscale image into a binary image, i.e. an image containing only black and white pixels. In this case, once the threshold is identified, it is used to separate the pixels with intensities above and below the threshold value, generating a new layer of true and false values. This results in an image where metals, mainly weld, are represented by white pixels, while everything else is represented by black pixels.

Figure 1 shows the histogram of 26 images of parts with arrays of weld spheres. Three threshold values are selected to demonstrate the result of their application on one sample image, which is shown in Fig. 2.

This binarization process facilitates the detection of missing solder spheres by simplifying the image and making metals easier to identify. After exploring different binarization values, a threshold of 100 was selected and this value is used in various stages of the algorithm.

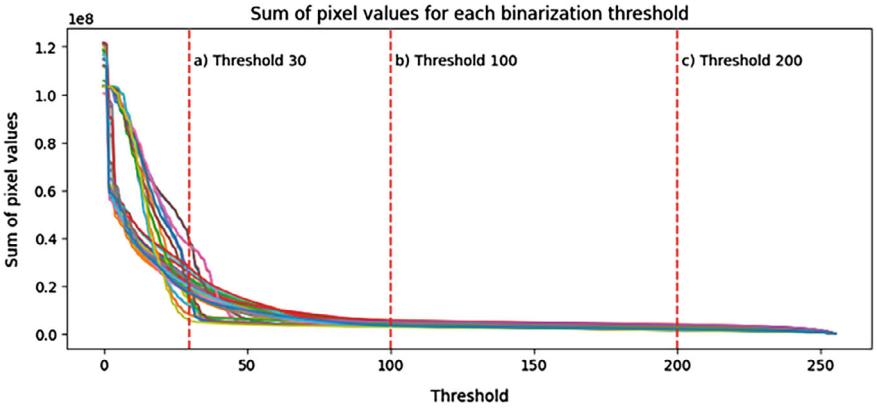


Fig. 1 Histogram of pixel values of 26 images of solder ball arrays

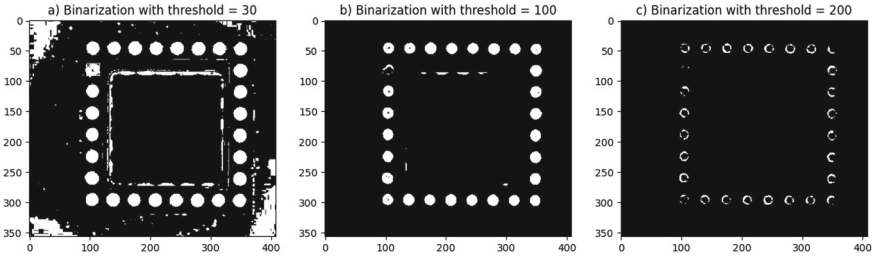


Fig. 2 Binarization of a sample image of a solder ball array with a threshold value of **a** 30, **b** 100, and **c** 200

3.2.2 Ball Detection

Once the image is binarized, the pixels are grouped into sets to create a list of shapes, whose characteristics may vary. For each shape or group of pixels in the list, the following metrics are calculated:

- **Area**: using the `cv2.contourArea` function.
- **Perimeter**: Using the `cv2.arcLength` function.
- **Circularity**: Computed with the formula:

$$\text{Circularity} = \frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2}, \tag{1}$$

where values close to 1 indicates how similar is a contour with respect to a prefect circle.

For this case, if the circularity of the outline is between 0.3 and 1.2, the outline is considered to be sufficiently circular and we proceed to extract more features, such as the position of the bounding rectangle using the Python function `cv2.boundingRect`. This function returns the (X) and (Y) coordinates of the upper left corner of the rectangle, as well as its width (W) and height (H). With these measurements, the center of the bounding rectangle and its area are calculated, adding all these features to the feature list. Then, those rectangles whose features are outliers are filtered out. In this way, the spheres that will be taken into account will be those with similar measures, following the next steps:

1. **Delimit area of highest concentration**: The first (Q1) and third quartile (Q3) of the area of the rectangle delimiting the contours and the interquartile range (IQR) are calculated. Then, the mean and standard deviation of this area are calculated. With these values, contours whose bounding rectangle area is within a range defined by the mean ± 1.5 times the standard deviation are filtered out. This process helps to eliminate contours that can be considered outliers in terms of their area.
2. **Filter by coordinate (X) of the center of the bounding rectangle (cx)**: The above process is repeated, but this time for the (X) coordinate of the center of the bounding rectangle. This helps to eliminate contours that can be considered outliers in terms of their horizontal position.
3. **Filter by the (Y) coordinate of the center of the bounding rectangle (cy)**: The process is repeated once again, but now for the (Y) coordinate of the center of the bounding rectangle. This helps to eliminate contours that can be considered outliers in terms of their vertical position.
4. **Create an array with the coordinates (cx) and (cy)**: Once the contours have been filtered, an array is created with the coordinates of the center of the bounding rectangles of the remaining contours.
5. **Calculate the distance matrix**: The distance matrix between all coordinates is calculated using the `pdist` function of SciPy library. This matrix contains the distance between each pair of contours.

6. **Find the minimum distance for each point**: Fill the diagonal of the distance matrix with inf to avoid getting zero when looking for the minimum. Then, find the minimum distance for each contour by finding the minimum value in each row of the distance matrix.
7. **Calculate the average distance and radius**: Calculate the average distance of the minimum distances and divide this value by 2 to obtain the radius.

Finally, the positions of the (X) and (Y) centers (cx) and (cy) are used to calculate the slopes of the linear regression lines that can be formed with those points and identify the direction of the part on each axis. In this way, a grid of the part can be generated, identifying the areas where a weld sphere should be present. Figure 3(top) shows the result of the algorithm described above for a straight front view of a sample part, while Fig. 3(bottom) depicts the result for a tilted front view of a sample part; both showing that the location of the spheres is correctly estimated by the proposed segmentation procedure.

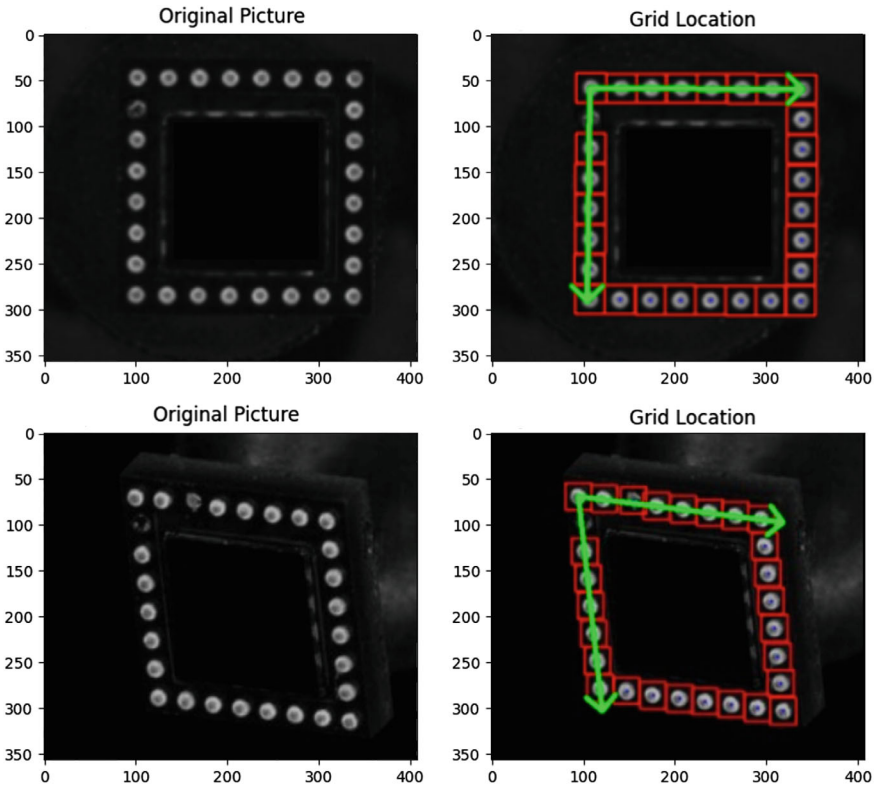


Fig. 3 Location of spheres and the orientation of the array in an image with straight front view (top) and tilted front view (bottom) of the part

3.3 Dataset Creation

The process of sphere identification is crucial for the dataset creation and training of the machine learning models. The regions of interest, which are segments of the original image, are automatically generated and then subjected to manual classification. This classification is performed by a semiconductor inspection expert, who labels each image segment according to the state of the solder sphere it contains.

3.3.1 Manual Classification

For the manual classification phase, a total of 3714 image clippings were generated and classified by a semiconductor inspection expert. Of these, 1330 correspond to weld spheres in good condition, classified as 'ball'. In addition, 95 cases of missing or fallen solder spheres were identified, classified as 'mb'. The rest of the cuttings, a total of 2289, correspond to other parts of the part and were classified as 'no_ball'.

It is important to note that two sets of initial images were created. The first, called 'dots', is composed of clippings that are delimited exclusively to the weld spheres, as shown in Fig. 4. The second set, called 'meanDist', includes snippets generated by taking into account an average distance between each sphere identified in the images, as shown in Fig. 5. This approach allows visualizing if there is any other weld sphere in the vicinity of the target area, providing valuable information to the models about the possible presence of a sphere at that position.

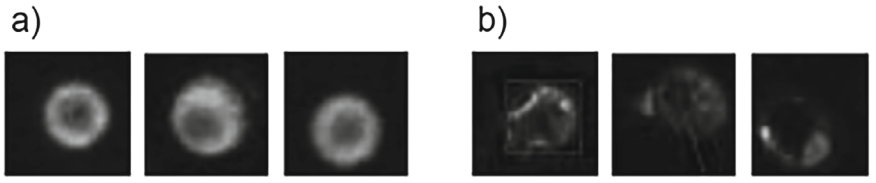


Fig. 4 Example images of the 'dots' set, where **a** shows 3 samples from the 'ball' class and **b** shows 3 samples from the 'mb' class

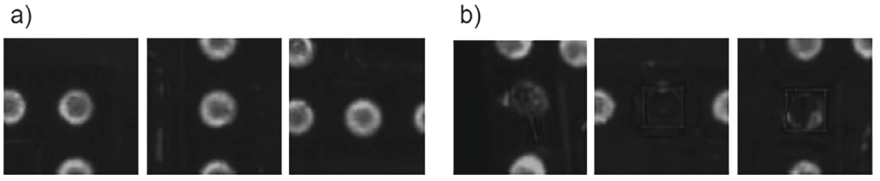


Fig. 5 Example images of the 'meanDist' set, where **a** shows 3 samples from the 'ball' class and **b** shows 3 samples from the 'mb' class

3.3.2 Feature Enhancement

Feature enhancement or intensification can significantly improve recognition in any machine learning task. This feature intensification process takes the original images to create a new enhanced dataset. In particular, the RGB channels are used to store information related to some features of the original images. The feature enhancement procedure is detailed below:

1. **Conversion to grayscale**: If the image is not in grayscale, it is converted using the `cv2.cvtColor` function.
2. **Binarization**: A threshold is applied to the grayscale image to create a binary image. Pixels with an intensity greater than 100 (value identified in the analysis in Sect. 3.2.1) are considered true (white), while those with a lower intensity are considered false (black).
3. **Edge detection**: The Canny method is used to detect edges in the binarized image, generating an edge image called `edges`.
4. **Creating a black image**: A black image of the same size as the original image is created.
5. **Normalization**: Images are converted to `float32` and normalized to the range `[0,1]`.
6. **Combination of images in different channels**: Depending on the number of features to be considered, the RGB channels are used as follows to form a new image:
   (a) 1 feature: the normalized image is combined with two black images, as shown in Fig. 6 (top).
   (b) 2 features: the normalized image is combined with the edge image and a black image, as shown in Fig. 6 (middle).
   (c) 3 features: the normalized image, the edge image and the binarized image are combined, as shown in Fig. 6 (bottom).

3.3.3 Oversampling

Since our data set was highly unbalanced, with 1330 images from the ‘ball’ class and only 95 from the ‘mb’ class, it was necessary to apply an oversampling to balance the classes. Oversampling is a technique commonly used in machine learning to increase the number of samples in the minority class, in this case, the ‘mb’ class. To achieve balance, the images of the ‘mb’ were copied until the same number of samples as the ‘ball’ class was reached.

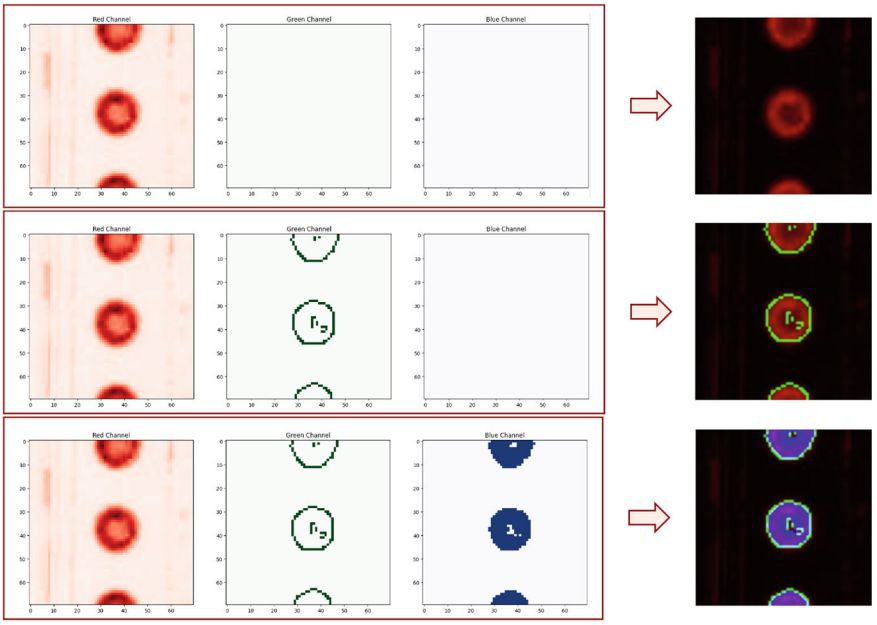


Fig. 6 Display of the RGB channels of a sample after the feature enhancement procedure for 1 feature (top), 2 features (middle), and 3 features (bottom)

3.3.4 Training and Test Datasets

After oversampling, the data set was divided into two parts: 80% for training and 20% for validation. For the training set, data augmentation techniques were applied to increase the diversity of the samples and improve the model's ability to generalize. These techniques included:

- **Rotation**: Images were randomly rotated within a range of 20° to recognize objects from different angles.
- **Scale shift**: A random scale change was applied on both the horizontal and vertical axes within a range of 0.2, allowing the model to recognize objects of different sizes.
- **Horizontal flip**: Images were randomly flipped on the horizontal axis, to recognize objects in different orientations.
- **Pixel fill**: For pixels that remained outside the input boundaries after rotation or scale change, the 'nearest' fill strategy was used, which fills the pixels with the nearest pixel value.

In addition, the image data generator `ImageDataGenerator` from Keras was used to preprocess the images. This generator not only performs data augmentation, but also rescales the images by dividing them by 255 to convert the pixel values into a range of 0 to 1, which is more suitable for deep learning models.

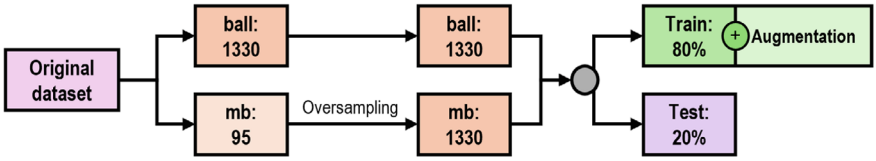


Fig. 7 Class leveling and splitting of training and test datasets

The image data generator also automatically divides the data into training and validation sets, using 20% of the data for validation. This data is used to evaluate the model performance during training, as described in Fig. 7.

Finally, the `flow_from_directory` method was used to load the images from the training directory. This method generates batches of image tensor data with class labels in real time. The images are resized to the specified dimensions and generated in batches of the specified size (batch_size). The class labels are encoded in categorical format, suitable for multi-class classification.

3.4 Classification Models

Two pretrained models, VGG16 and MobileNet, are used in addition to a custom model designed specifically for this task.

The pretrained models are used to take advantage of features that have already been learned with other datasets. However, to adapt these models to our specific problem, the last layer of the pretrained models is unfrozen and trained with our dataset. This process, known as fine-tuning, allows the pretrained models to better fit our dataset. The characteristics of the three models are described below:

1. **VGG16**: This model uses the pretrained architecture from ImageNet as its basis. It is a convolutional neural network consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. In this model, the fully connected layers of the VGG16 architecture are omitted, and the following layers are added:

   - A **Flatten** layer to flatten the output of the base model.
   - A **Dense** layer or fully connected layer with 512 neurons and ReLU activation function, regularized with L2 and weights initialized using the `HeNormal` initializer.
   - A **Dropout** layer with a 50% rate to regularize the model and reduce overfitting.
   - A **Dense** layer with a number of neurons equal to the number of classes and Softmax activation function for classification, also regularized with L2 and with weights initialized using the `GlorotNormal` initializer.

   The final architecture of the modified VGG16 model is shown in Fig. 8.

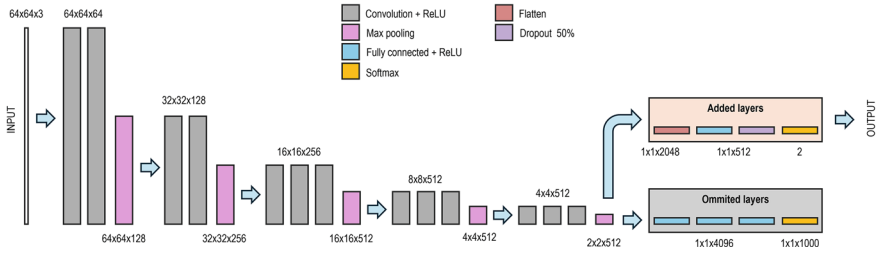


Fig. 8 Modified VGG16 architecture with added layers

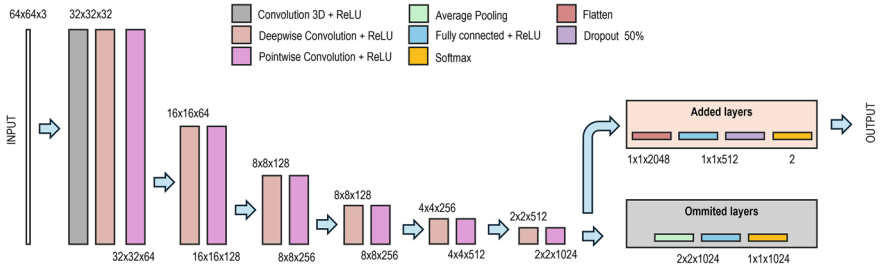


Fig. 9 Modified MobileNet V1 architecture with added layers

2. **MobileNet**: This model also uses the pretrained architecture from ImageNet as its basis. MobileNet is an efficient convolutional neural network designed for mobile devices and computer vision applications at the edge. In this model, the fully connected layers of the MobileNet architecture are omitted, and the same layers are added as with previous model. The final architecture of the modified MobileNet model is shown in Fig. 9
3. **Customized model**: This model consists of the following layers:

   - Three blocks of **Conv2D** and **MaxPooling2D** layers. The Conv2D layers have 32, 64 and 128 filters respectively and use the ReLU trigger function. MaxPooling2D layers are used to reduce the spatial dimensionality of the convolutional layer output.
   - A **Flatten** layer to flatten the exit of the last block.
   - Two **Dense** layers with 128 and 64 neurons respectively and ReLU activation functions, regularized with L2 and with weights initialized using the `GlorotNormal` initializer.
   - Two **Dropout** layers after each Dense layer to regularize the model and reduce overfitting.
   - A **Dense** layer with a number of neurons equal to the number of classes and Softmax activation function for classification, also regularized with L2 and with weights initialized using `GlorotNormal`.

   The final architecture of the new custom model is shown in Fig. 10.

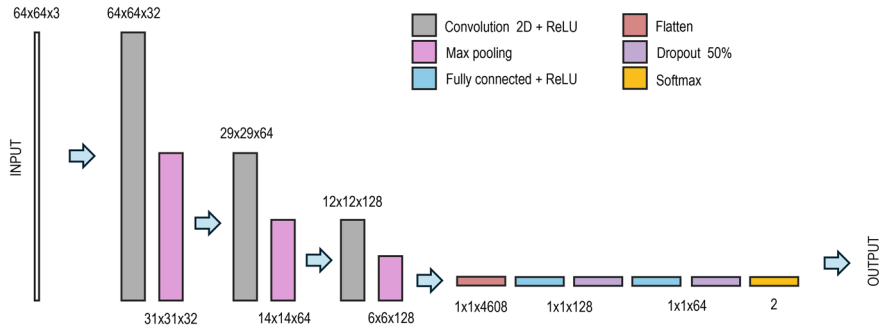


Fig. 10 Proposed convolutional custom model architecture

3.5 Hyperparameters

Hyperparameters are parameters that are not learned during model training, but are set prior to training. These may include learning rate, batch size, and number of training epochs, etc. A grid search is performed to find the best hyperparameters for each model. Grid search involves trying different combinations of hyperparameters and selecting the combination that produces the best model performance.

The hyperparameters being tuned include image size, batch size, number of epochs, and learning rate; and the values being tested are given below:

- **Image size**: (32, 32) and (64, 64) pixels.
- **Batch size**: 16, 32 and 64 images.
- **Number of epochs**: set to a fixed value 50, but the Early Stopping function is used to stop training if the validation accuracy does not improve after 15 consecutive epochs.
- **Learning rate**: 0.1, 0.01, 0.001 and 0.0001 rates.

Finally, the following configurations are considered as additional hyperparameters:

- **Crop**: the crop size of the images 'dots' and 'meanDist' described in Sect. 3.3.1.
- **Features**: the enhanced image features used, with '0' as the original monochromatic image with the same values in the RGB channels, and '1', '2' and '3' corresponding to the number of features stored in the RGB channels, as described in Sect. 3.3.2.
- **Model**: the models under evaluation 'mobilenet', 'vgg16' and a 'custom' described in Sect. 3.4.

This result in a total of 576 hyperparameter combinations that must be evaluated, as shown in Fig. 11. Each of these models is trained with two classes, 'ball' and 'mb'.

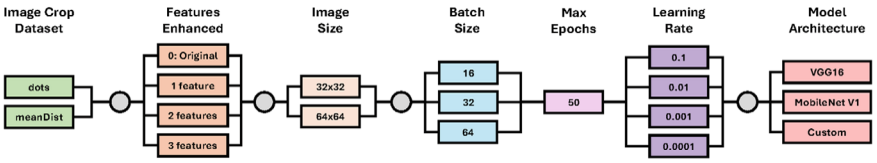


Fig. 11 Hyperparameters configurations for model training

3.6 Model Evaluation

Once the models have been trained, their performance is evaluated in order to compare them effectively. The evaluation is performed as follows:

1. **Evaluation on the Test Set**: Each model is evaluated using a data set that has not been used during training, known as the test set.
2. **Training and Testing Times**: Training and testing times are recorded for comparison purposes.
3. **Classification Metrics**: Various classification metrics are calculated, such as accuracy, recall, specificity and F1 value. These metrics provide a more detailed view of model performance than accuracy alone.
4. **Model Comparison**: Finally, models are compared with each other based on classification metrics and error analysis.

4 Results

This section show the results of the implementation of the models described before. First, a performance evaluation was carried out to asses the training and testing times of the introduced models. After this, the classification results are given for each model configuration.

4.1 Performance Evaluation

The evaluation of the proposed models was performed in a MSI Katana GF76 laptop, with a Intel Core i7-11800H CPU, a Nvidia GeForce RTX 3060 GPU, 16 GB of RAM and 1 TB of SSD storage.

In order to evaluate the training and testing times for each model, we started the performance evaluation using the 576 hyperparameter configurations with only on CPU resource. For this performance evaluation, the number of images per class was limited to 100 for training and 1000 for testing.

It was obtained an average training time of 688 s and an average testing time of 4867 s per model configuration. This means a duration of 222 h for the 576 hyperparameter configurations with the CPU resource, equivalent to 9.25 days of training and testing.

TensorFlow functionalities were enabled to use the GPU during training and testing of the models, under the same 576 hyperparameter configurations. An average training time of 12 s and an average testing time of 47 s was obtained. This means a duration of 141.6 min of training and testing time for the 576 hyperparameter configurations, equivalent to 2.36 h.

Comparing the performance obtained with CPU and GPU, a reduction in training time of 97.09% and in testing time of 98.39% were observed.

4.2 Classification Results

The three classification models were evaluated under different hyperparameter settings: a modified MobileNet model, a modified VGG16 model and a custom model. The classification results were measured in terms of precision, accuracy, recall, and F1 score, as well as training and testing times were evaluated. The results for the 576 combinations of hyperparameter configurations were computed and the top 10 performances per model are given in Table 1.

Additionally, the comparison of the classification results are shown in Figs. 12, 13, 14, 15, 16, 17 and 18, where each figure shows the statistics of the classification metrics obtained for the 576 hyperparameter combinations, with respect to the variation of one specific hyperparameter configuration.

Specifically, Fig. 12 shows the statistics of the classification metrics with respect to the model used, Fig. 13 shows the statistics of the classification metrics with respect to the image crop configuration, Fig. 14 shows the statistics of the classification metrics with respect to the image size, Fig. 15 shows the statistics of the classification metrics with respect to the number of features used, Fig. 16 shows the statistics of the classification metrics with respect to the batch size, Fig. 17 shows the statistics of the classification metrics with respect to the elapsed epochs, and Fig. 18 shows the statistics of the classification metrics with respect to the learning rate used.

Finally, Figs. 19 and 20 show the results for the training and testing times with respect to the classification model and the image crop configuration, respectively.

Table 1 Top 10 performance results for each model with respect to the F1 score obtained for all combinations of hyperparameter configurations

	Model	Crop	Size	Ft	BS	E	LR	Acc	Prec	Rec	F1
1	custom	meanDist	(64, 64)	0	32	7	0.001	0.931	0.898	0.931	0.905
2	custom	meanDist	(64, 64)	0	32	18	0.0001	0.927	0.893	0.927	0.905
3	custom	meanDist	(64, 64)	0	64	38	0.0001	0.927	0.891	0.927	0.904
4	mobilenet	meanDist	(64, 64)	1	16	8	0.01	0.931	0.889	0.931	0.902
5	custom	meanDist	(32, 32)	1	16	15	0.001	0.923	0.889	0.923	0.902
6	custom	meanDist	(64, 64)	1	16	26	0.01	0.924	0.888	0.924	0.902
7	custom	meanDist	(32, 32)	0	16	15	0.001	0.932	0.888	0.932	0.902
8	custom	meanDist	(64, 64)	0	16	10	0.001	0.919	0.889	0.919	0.902
9	vgg16	meanDist	(64, 64)	0	64	1	0.0001	0.933	0.871	0.933	0.901
10	vgg16	meanDist	(64, 64)	0	32	1	0.1	0.933	0.871	0.933	0.901
11	vgg16	meanDist	(64, 64)	0	64	1	0.1	0.933	0.871	0.933	0.901
12	vgg16	meanDist	(64, 64)	0	64	1	0.001	0.933	0.871	0.933	0.901
13	vgg16	meanDist	(64, 64)	0	16	1	0.001	0.933	0.871	0.933	0.901
14	vgg16	meanDist	(64, 64)	0	16	1	0.1	0.933	0.871	0.933	0.901
15	vgg16	meanDist	(32, 32)	0	32	1	0.1	0.933	0.871	0.933	0.901
16	vgg16	meanDist	(32, 32)	0	16	1	0.01	0.933	0.871	0.933	0.901
17	vgg16	meanDist	(64, 64)	1	32	1	0.001	0.933	0.871	0.933	0.901
18	vgg16	meanDist	(64, 64)	1	16	1	0.001	0.933	0.871	0.933	0.901
19	mobilenet	meanDist	(64, 64)	0	64	12	0.1	0.933	0.871	0.933	0.901
20	mobilenet	meanDist	(64, 64)	1	32	10	0.01	0.933	0.871	0.933	0.901
21	mobilenet	meanDist	(64, 64)	3	16	12	0.01	0.933	0.871	0.933	0.901
22	mobilenet	meanDist	(64, 64)	2	32	33	0.001	0.933	0.871	0.933	0.901
23	mobilenet	meanDist	(64, 64)	3	16	2	0.001	0.933	0.871	0.933	0.901
24	mobilenet	meanDist	(64, 64)	0	16	15	0.1	0.933	0.871	0.933	0.901
25	mobilenet	meanDist	(64, 64)	1	16	14	0.001	0.933	0.871	0.933	0.901
26	mobilenet	meanDist	(64, 64)	3	32	19	0.01	0.933	0.871	0.933	0.901
27	mobilenet	meanDist	(64, 64)	2	32	13	0.01	0.933	0.871	0.933	0.901
28	custom	meanDist	(64, 64)	0	16	1	0.01	0.933	0.871	0.933	0.901
29	custom	meanDist	(32, 32)	0	32	34	0.001	0.933	0.871	0.933	0.901
30	custom	meanDist	(64, 64)	0	64	1	0.01	0.933	0.871	0.933	0.901

Ft = Features; BS = Batch Size; E = Epochs; LR = Learning Rate

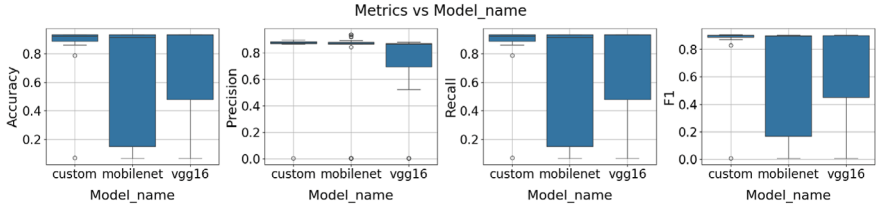


Fig. 12 Comparative boxplot showing the classification metrics with respect to the classification model

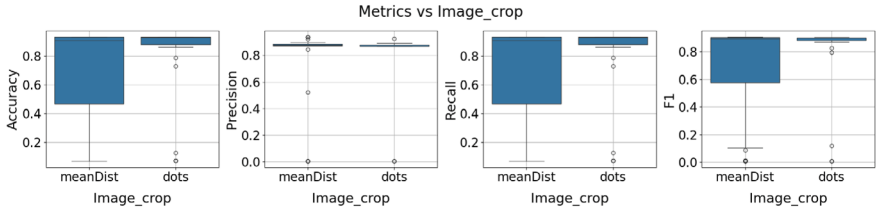


Fig. 13 Comparative boxplot showing the classification metrics with respect to the image crop

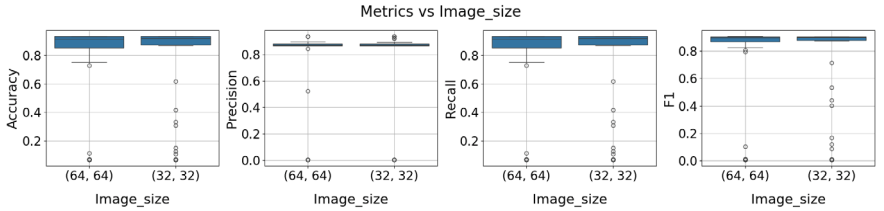


Fig. 14 Comparative boxplot showing the classification metrics with respect to the image size

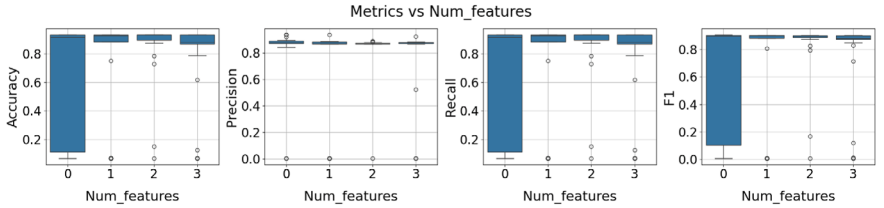


Fig. 15 Comparative boxplot showing the classification metrics with respect to the features

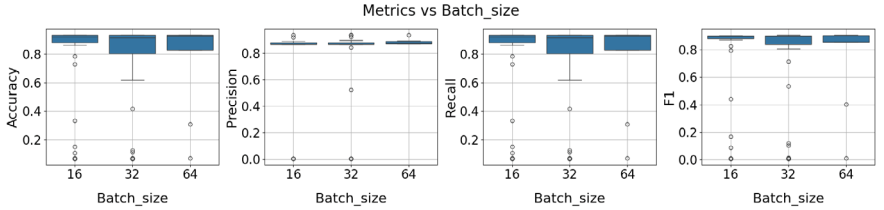


Fig. 16 Comparative boxplot showing the classification metrics with respect to batch size

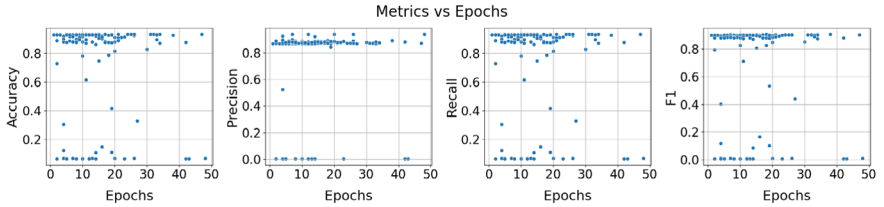


Fig. 17 Comparative boxplot showing the classification metrics with respect to the elapsed epochs

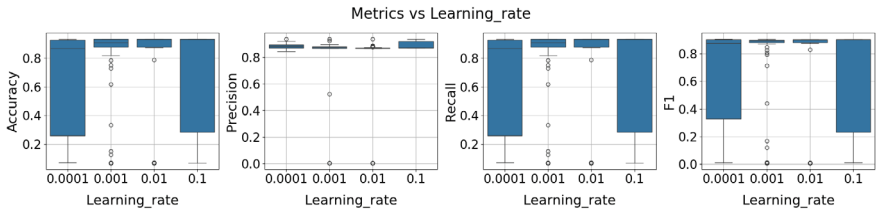


Fig. 18 Comparative boxplot showing the classification metrics with respect to the learning rate

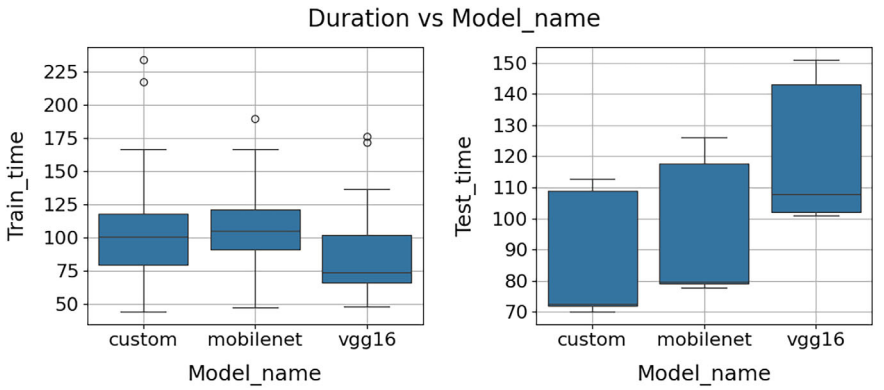


Fig. 19 Comparative boxplot showing the training and testing times with respect to the classification model

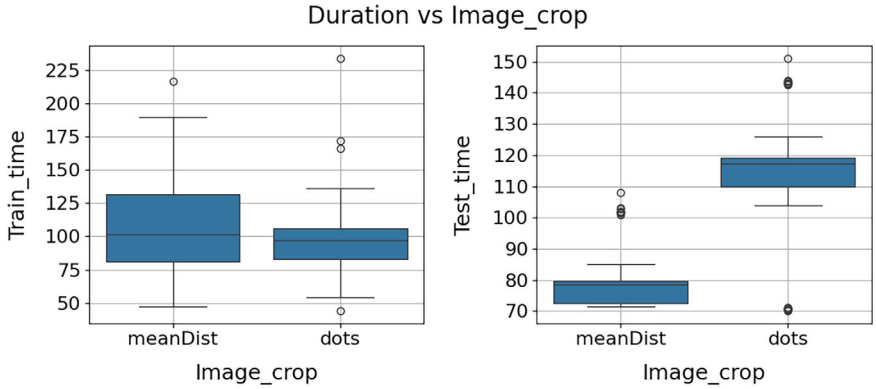


Fig. 20 Comparative boxplot showing the training and testing times with respect to the image crop configuration

5 Discussion

The experiments conducted in this paper show that the pretrained MobileNet and VGG16 models achieved a consistent accuracy of 93.3% in multiple configurations. The customized model also showed robust performance, achieving an accuracy of up to 93.3% in some configurations, as can be seen in Table 1, which shows the top 10 performances per model for all configurations, with respect to F1 score. It is worth noting that we considered F1 score as the reference metric for this dataset, since it is a highly unbalanced dataset between its two classes. In this aspect, the customized model presented better results, reaching up to 90.5%.

Different datasets can lead to important variations in precision, accuracy, recall and F1 score. For example, models trained on the meanDist images generally achieved better accuracy, up to 93.3%, while models trained on the dots images also performed well, but with slightly lower performance in some cases. Therefore, it is possible to state that the type of image snipping can influence how the model generalizes to new data and datasets with more diverse and representative samples can help the model perform better on unseen data. Additionally, it is well known that the nature of the dataset affects the features that the model learns. For example, the meanDist images might emphasize different features compared to the dots dataset, leading to variations in model performance.

Regarding the image size, both 32 x 32 and 64 x 64 sizes produced similar levels of accuracy. However, it was observed that training and testing times varied, with larger image sizes requiring more computational resources, while larger batch sizes (64) resulted in longer training times, but did not necessarily improve accuracy. Optimal batch sizes were 16 or 32 for most the best performed configurations, as can be observed in Table 1.

On the other hand, it can be seen that increasing the number of epochs did not always lead to higher accuracy. For example, MobileNet with 8 epochs and a learning

rate of 0.01 achieved an accuracy of 93.1%. It is also worth noting that a learning rate of 0.1 was effective for MobileNet and VGG16, while the customized model performed better with learning rates of 0.001 or 0.0001.

Finally, as for training times, these ranged from approximately 34 to over 216 s, while testing times ranged between 72 and 104 s. However, models with longer training times did not always result in higher accuracy and F1 score. In particular, models trained on the dots images had longer testing times compared to those trained on the meanDist images, which obtained the best results.

6 Conclusions

This study included the design and evaluation of three classification models for the detection of missing solder balls in encapsulated semiconductors. It was observed that the classification models achieved high accuracy even with few training epochs. This may occur because MobileNet and VGG16 were pretrained on similar datasets, enabling them to perform accurate classifications without extensive additional training. It was also shown that, although the best classification results were obtained with the original monochromatic images, the feature enhancing technique provided a significant improvement of the classification robustness, this by including the edged and thresholded version of the original images as additional information in the RGB channels.

Additionally, the image cropping technique proved to be valuable for enhancing the accuracy and robustness of the classification models by including more information of the relevant features of the images. Proper adjustment of cropping parameters is essential to maximize benefits without compromising the integrity of the information. By removing irrelevant details and emphasizing important features, image cropping can significantly improve classification accuracy. However, it is crucial to select cropping parameters carefully to avoid losing relevant information that could negatively impact model performance.

Also, weight initialization plays an important role during training. In some cases, the initial model weights may be set in a way that already provides good accuracy on the specific dataset. This highlights the importance of choosing an appropriate initialization strategy to ensure optimal model performance from the start of training.

As future work, it is desired to explore the implementation of models that can infer whether there is a missing ball from the picture of the entire component, without the need for the segmentation procedure. Improving the dataset by increasing the number of samples of the missing ball class is also considered. Future work will also address automatic detection of the threshold level and the real-time implementation of the best performing models. Finally, it is also desired to explore the application of the quantum Fourier transform for the detection of solder ball defects in encapsulated semiconductors.

References

1. IC Insights, Inc.: The 2022 mcclean report: A complete analysis and forecast of the semiconductor industry. Technical report, IC Insights, Inc. (February 2022). Research Bulletin, February 10, 2022. https://www.icinsights.com/files/data/articles/documents/1430.pdf
2. PR Newswire: Semiconductor market - 2022 to 2027: The increasing number of data centers drives market growth. Technical report, PR Newswire (September 2023). Published: September 18, 2023. https://www.prnewswire.com/news-releases/semiconductor-market---2022-to-2027--the-increasing-number-of-data-centers-drives-market-growth-301930281.html
3. Technavio Research: Trends in semiconductor market driven by iot and data centers. Technical report, Technavio (September 2023). Published: September 18, 2023. https://kpmg.com/kpmg-us/content/dam/kpmg/pdf/2024/global-semiconductor-industry-outlook.pdf
4. Silva, C.N., Ferreira, N.P., Meireles, S.S., Otani, M., Silva, V.J., Freitas, C.A.O., Oliveira, F.G.: The visual inspection of solder balls in semiconductor encapsulation. In Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO) 1, 750–757 (2022)
5. Chang, Y.-M., Wei, C.-C., Chen, J., Hsieh, P.: Classification of solder joints via automatic mistake reduction system for improvement of aoi inspection. In: 2018 13th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), pp. 150–153 (2018)
6. Schlosser, T., Friedrich, M., Beuth, F., Kowerko, D.: Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks. Journal of Intelligent Manufacturing 33, 1099–1123 (2022). https://doi.org/10.1007/s10845-021-01906-9
7. Mordor Intelligence: Semiconductor Metrology and Inspection System Market. Accessed: 2024-09-09 (2022). https://www.mordorintelligence.com/industry-reports/semiconductor-inspection-equipment-market
8. Zhang, G., Cui, K., Hung, T.-Y., Lu, S.: Defect-gan: High-fidelity defect synthesis for automated defect inspection. Proc. of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) (2021)
9. López-de-la-Rosa, F., Gómez-Sirvent, J.L., Morales, R., Sánchez-Reolid, R., Fernández-Caballero, A.: Defect detection and classification on semiconductor wafers using two-stage geometric transformation-based data augmentation and squeezenet lightweight convolutional neural network. Computers & Industrial Engineering 180, 109549 (2023). https://doi.org/10.1016/j.cie.2023.109549
10. Kong, D., Hu, X., Gong, Z., Zhang, D.: Segmentation of void defects in x-ray images of chip solder joints based on pcb-deeplabv3 algorithm. Scientific Reports 14, 11925 (2024). https://doi.org/10.1038/s41598-024-61643-w
11. Chien, J.-C., Wu, M.-T., Lee, J.-D.: Inspection and classification of semiconductor wafer surface defects using cnn deep learning networks. Applied Sciences 10(15), 5340 (2020). https://doi.org/10.3390/app10155340
12. Zhang, Q., Zhang, M., Gamanayake, C., Yuen, C., Geng, Z., Jayasekara, H., Woo, C.-w., Low, J., Liu, X., Guan, Y.L.: Deep learning based solder joint defect detection on industrial printed circuit board x-ray images. Complex & Intelligent Systems 8, 1525–1537 (2022). https://doi.org/10.1007/s40747-021-00600-w
13. Rosa, F.L., Sánchez-Reolid, R., Gómez-Sirvent, J.L., Morales, R., Fernández-Caballero, A.: A review on machine and deep learning for semiconductor defect classification in scanning electron microscope images. Applied Sciences 11(20), 9508 (2021). https://doi.org/10.3390/app11209508
14. Ma, J., Zhang, T., Yang, C., Cao, Y., Xie, L., Tian, H., Li, X.: Review of wafer surface defect detection methods. Electronics 12(8), 1787 (2023). https://doi.org/10.3390/electronics12081787

15. Gómez-Sirvent, J.L., López de la Rosa, F., Sánchez-Reolid, R., Morales, R., Fernández-Caballero, A.: Defect classification on semiconductor wafers using fisher vector and visual vocabularies coding. Measurement 202, 111872 (2022). https://doi.org/10.1016/j.measurement.2022.111872
16. Ren, Z., Fang, F., Yan, N., Wu, Y.: State of the art in defect detection based on machine vision. International Journal of Precision Engineering and Manufacturing-Green Technology 9, 661–691 (2022). https://doi.org/10.1007/s40684-021-00343-6
17. DeRose, J., Barbero, D.R.: Rapid Semiconductor Inspection with Microscope Contrast Methods (2023). https://www.leica-microsystems.com/science-lab/applied/rapid-semiconductor-inspection-with-microscope-contrast-methods/
18. Yu, J., Han, S., Lee, C.-O.: Defect inspection in semiconductor images using fast-mcd method and neural network. The International Journal of Advanced Manufacturing Technology 129, 1547–1565 (2023). https://doi.org/10.1007/s00170-023-12287-z

Membrane Pseudo-Bacterial Potential Field with GPU Acceleration for Mobile Robot Path Planning

Ulises Orozco-Rosas, Kenia Picos, and Oscar Montiel

Abstract Path planning is one of the most important tasks in mobile robots. This task is computationally intensive, and it is intensified by the complexity of the environment. This work presents the membrane pseudo-bacterial potential field algorithm with GPU (graphics processing unit) acceleration for mobile robot path planning. This proposal includes the combination of membrane computing, the pseudo-bacterial genetic algorithm, and the artificial potential field method. In this work, we will focus on testing the proposed algorithm in four test environments with distinctive challenges for each one. These experiments are carried out to test the membrane pseudo-bacterial potential field algorithm in terms of path length and computation time. We show how data-intensive tasks in mobile robots can be processed efficiently using the GPU through a parallel computing implementation, in which different paths are concurrently calculated to select the best one. Experiments and simulation results are provided to show the effectiveness of the proposal in computational performance achieving a factor above 8× on CPU (central processing unit) and above 133× on GPU.

Keywords Membrane computing · Pseudo-bacterial genetic algorithm · Artificial potential field · GPU acceleration · Path planning · Mobile robots

U. Orozco-Rosas (✉) · K. Picos
CETYS Universidad, Tijuana, B.C., Mexico
e-mail: ulises.orozco@cetys.mx

K. Picos
e-mail: kenia.picos@cetys.mx

O. Montiel
Instituto Politécnico Nacional, CITEDI-IPN, Tijuana, B.C., Mexico
e-mail: oross@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_8

1 Introduction

Path planning is one of the most fundamental tasks in autonomous mobile robots. As stated in [1], the capability of effectively planning a robot's motions is "eminently necessary since, by definition, a robot accomplishes a task by moving in the real world". Nowadays, mobile robots have become increasingly autonomous in activities with greater effectiveness over time. In that sense, autonomous robots require efficient complex algorithms to solve navigation and path planning problems [2]. The resultant path must be collision-free and satisfy certain optimization criteria [3]. In this work, the path planning problem involves generating a feasible route (navigation path) to drive the robot from the start position to the goal position in a static environment (robot workspace) with obstacles without colliding with them and optimizing the path concerning its length. Figure 1 depicts the path planning problem.

This work aims to implement the membrane pseudo-bacterial potential field algorithm [4] on a graphics processing unit (GPU) to accelerate the path-planning process through parallel computation [5]. The main objective of parallel computing is to speed up the computation process. From a pure calculation perspective, parallel computing can be stated as a form of computation in which many calculations are carried out simultaneously, operating on the principle that large computation problems can often be divided into smaller ones, which are then solved concurrently [6].

The organization of this work is as follows: Sect. 2 presents the fundamentals of the membrane pseudo-bacterial potential field algorithm and a concise description of its main components. These components are membrane computing [7], the pseudo-bacterial genetic algorithm [8], and the artificial potential field method [9]. Section 3

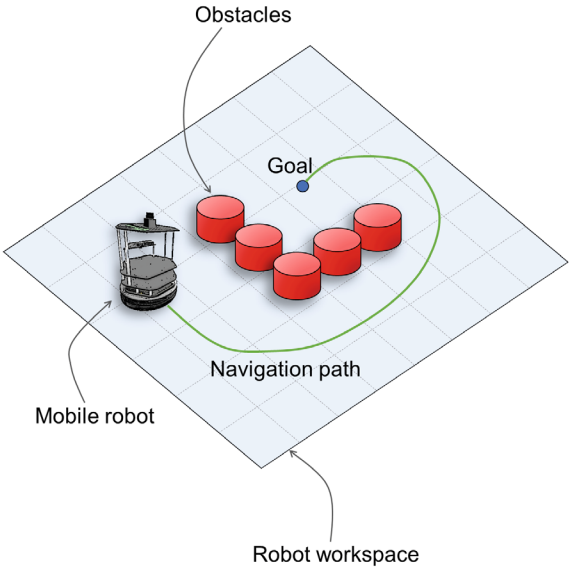


Fig. 1 Mobile robot path planning in a static environment

describes the membrane pseudo-bacterial potential field algorithm for mobile robot path planning. In Sect. 3.1, the membrane pseudo-bacterial potential field algorithm is explained, where the fitness function based on the artificial potential field method represents the most expensive process in terms of computation time. Section 3.2 describes the artificial potential field method that acts as a fitness function and as a local planning method for the membrane pseudo-bacterial potential field algorithm. Section 4 presents the results from the experiments and the performance evaluation. The experimental framework consists of four test environments. They were designed to demonstrate the relevance of the membrane pseudo-bacterial potential field algorithm implemented on a GPU to accelerate the path planning computation. Finally, in Sect. 5 the conclusions are given.

2 Fundamentals

The membrane pseudo-bacterial potential field algorithm employs the artificial potential field method and membrane computing, using a metaheuristic based on a pseudo-bacterial genetic algorithm as the global optimization method to solve efficiently path planning problems in autonomous mobile robot navigation. The path planning computation is accelerated by employing GPU-based parallel computing techniques.

2.1 Membrane Computing

Membrane computing, called P systems or membrane systems is part of natural computing [10]. It was initiated by Gheorghe Păun inspired by the compartmentalized structure and interactions of living cells [7]. The resultant membrane computing models are distributed parallel schemes that evolve through rules and process the multisets of objects into compartments that are hierarchically defined [11]. Nowadays, membrane computing has been used to solve complex optimization problems such as TSP (traveling salesman problem) [12], maximum clique problem [13], minimum vertex cover problem [14], among others. Current contributions of membrane computing have given good results in terms of optimization even when combined with neural networks or evolutionary algorithms [12, 15].

Membrane computing or P systems consist of membrane structures with objects in their membranes, with specific evolution rules like transformation and communication to merge and divide membranes [7]. P systems are classified into three main types, cell-like P systems that contain one-membrane cell, tissue-like P systems consisting of several one-membrane cells in a common environment, and neural-like P systems that consider neurons as their cell [16].

In this work, a cell-like P system is the main component of the membrane pseudo-bacterial potential field algorithm since it is simple and practical to implement with

parallel computing. Cell-like P systems present a hierarchical structure of membranes, types of rules, e.g., transformation and communication, and inherent parallelism [4]. These characteristics and strengths are desirable from a computational point of view and attractive for modeling complex problems [17]. Figure 2 shows the membrane pseudo-bacterial potential field structure.

In that way, the membrane pseudo-bacterial potential field algorithm for mobile robot path planning is composed of a cell-like P system that evolves the set of parameters k_{rep} and k_{att} required to generate the path with the artificial potential field method. Therefore, the membrane pseudo-bacterial potential field algorithm employs a dynamic structure with active membranes, specifically a one-level membrane structure with rules, such as membrane merger and division [18, 19]. The membrane merger is helpful for information communication enhancement among individuals, i.e., the set of parameters, k_{rep} and k_{att}, and the membrane division is beneficial to improve the search capability [20].

2.2 *Pseudo-Bacterial Genetic Algorithm*

The evolutionary algorithms are generic population-based metaheuristic optimization algorithms inspired by biological evolution, such as selection, reproduction,

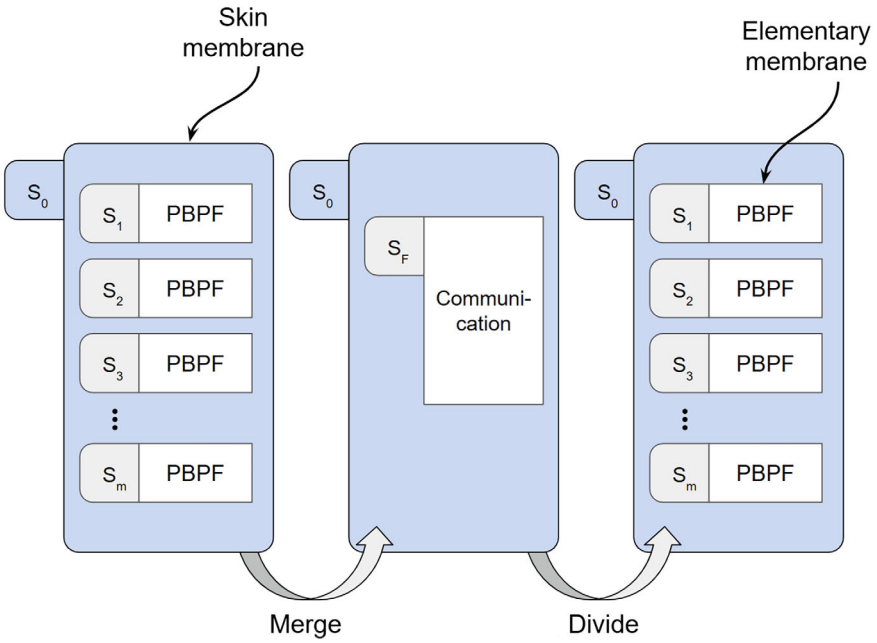


Fig. 2 Membrane pseudo-bacterial potential field structure

mutation, and recombination. The candidate solutions in the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions.

Nawa et al. proposed an evolutionary algorithm called the pseudo-bacterial genetic algorithm, which was successfully applied to extract rules from input and output data [21]. The pseudo-bacterial genetic algorithm has most of the advantages of a genetic algorithm and includes others that help to improve computational performance. The main process of the pseudo-bacterial genetic algorithm contains the bacterium which can carry a copy of a gene from a host cell and insert it into an infected cell. Hence, by this process called bacterial mutation, the characteristics of a single bacterium can be spread to the rest of the population, so this method mimics the process of microbial evolution [22].

The biological model of bacterial cells inspires the bacterial mutation operator. It means that the pseudo-bacterial genetic algorithm mimics the phenomenon of microbial evolution [23]. To find the global optimum, exploring different regions in the search space that the current population has not covered is necessary. This is achieved by adding new information to the bacteria; the data is generated randomly by the bacterial mutation operator applied to all the bacteria, one by one.

Figure 3 shows the bacterial mutation process [24]. First, some clones (copies) of the bacteria are created. Then, one segment with a certain length randomly selected undergoes mutation in each clone, but one clone is left unmutated. After mutation, the clones are evaluated using the fitness function. The resulting clone with the best evaluation transfers the undergone mutation segment to the other clones. These three operations (mutation clones, selection of the best clone, and transfer of the segment subjected to mutation) are repeated until each segment of the bacterium has been once subjected to mutation. Finally, the best bacterium is maintained, and the clones are removed [25].

The pseudo-bacterial genetic algorithm is an evolutionary algorithm that gives convergence and improvement in the results [23], without adverse effects in the exploration. In the most basic form, the pseudo-bacterial genetic algorithm is modeled for computer simulation employing the difference equation shown in Eq. (1). Where t represents the generation counter, and $P(t+1)$ is the new population of bacteria attained from the current population of bacteria $P(t)$ after being improved by random variation (bacterial mutation) v and selection s [26].

$$P(t+1) = s(v(P(t))) \tag{1}$$

2.3 *Artificial Potential Field Method*

The artificial potential field method establishes an attractive potential field around the goal position and a repulsive potential field around obstacles. The two potential fields, attractive and repulsive form the total potential field. Therefore, the artificial

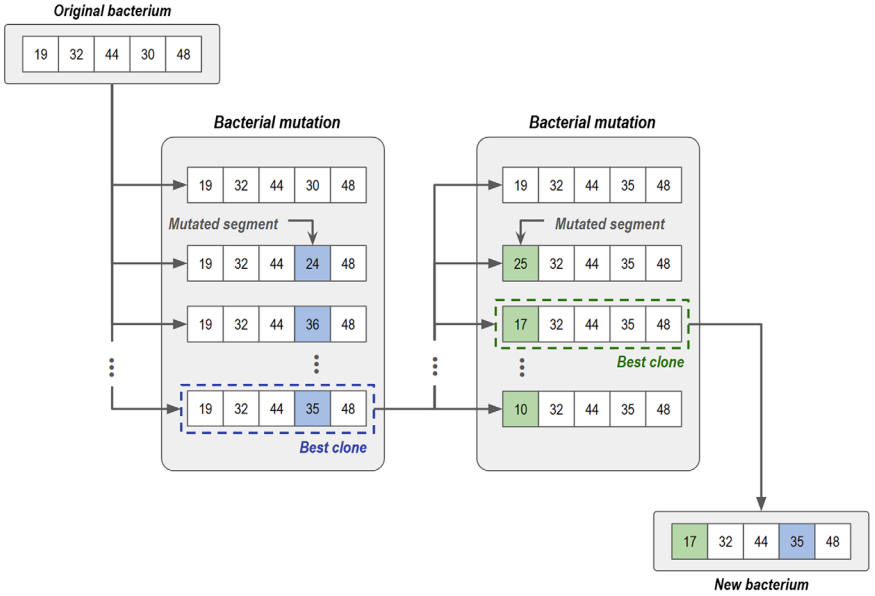


Fig. 3 The bacterial mutation process consists of three operations: mutation clones, selection of the best clone, and transfer of the segment subjected to mutation

potential field method can search the falling direction of the potential field to find a collision-free path, which is built from the start to the goal position [9].

In recent approaches, the artificial potential field method has been successfully used with other current methods to achieve more efficient path planning and to obtain better estimates for decision-making during navigation [27]. These approaches include reinforcement learning [28, 29], evolutionary algorithms [30], quantum computing [31], among others.

The artificial potential field method is essentially operated by the gradient descent method, which in each position of the mobile robot is directed toward minimizing the total potential field [32]. The goal position represents an attractive pole and the obstacles represent repulsive forces on the surface where the mobile robot will perform its movement [33, 34].

Equation (2) shows the total potential field denoted by $U(q)$. It is composed of the sum of the attractive potential field, $U_{att}(q)$, and the repulsive potential field, $U_{rep}(q)$.

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{2}$$

The attractive potential field is described by Eq. (3). Where q represents the mobile robot's position in a two-dimensional workspace. The vector q_f represents the goal position, and k_{att} is a positive scalar constant that represents the proportional gain

of the attractive potential field function. The expression $(q - q_f)$ is the distance between the current mobile robot and goal positions.

$$U_{att}(q) = \frac{1}{2} k_{att}(q - q_f)^2 \tag{3}$$

The repulsive potential field has a limited range of influence, this condition prevents the movement of the mobile robot from being affected by distant obstacles [35]. The repulsive potential field is described by Eq. (4). Where ρ_0 represents the limit distance of influence of the potential field, ρ is the shortest distance to the obstacle, and k_{rep} is a positive scalar constant that represents the proportional gain of the repulsive potential field function.

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_{rep}(\frac{1}{\rho} - \frac{1}{\rho_0})^2 & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \tag{4}$$

The generalized force $F(q)$ is obtained by the negative gradient of the total potential field, $U(q)$, and it is described by Eq. (5). The artificial potential field method requires the appropriate values of the proportional gains, k_{rep} and k_{att}, to guide the mobile robot to reach the goal position successfully [36].

$$F(q) = -\nabla U(q) \tag{5}$$

Therefore, the membrane pseudo-bacterial potential field algorithm with GPU acceleration is employed to search for the appropriate values of the proportional gains.

3 Membrane Pseudo-Bacterial Potential Field Algorithm for Mobile Robot Path Planning

This section explains the membrane pseudo-bacterial potential field (Algorithm 1) with GPU acceleration for mobile robot path planning. Where, the pseudo-bacterial potential field (PBPF) [24] is the main component of each elementary membrane as shown in Fig. 2. The pseudo-bacterial potential field comprises the pseudo-bacterial genetic algorithm and the artificial potential field method. Hence, the artificial potential field method (Algorithm 2) represents the evaluation function that is computed in parallel form regarding the number of available cores in the CPU or GPU as appropriate.

3.1 Membrane Pseudo-Bacterial Potential Field Algorithm

Algorithm 1 presents the pseudocode for the membrane pseudo-bacterial potential field algorithm that employs the following input parameters, start position q_0, goal

position q_f, and the environment configuration O_j. The environment configuration includes the obstacle position and its radius in format (x, y, r_o).

Algorithm 1 Membrane pseudo-bacterial potential field algorithm

Require: start position q_0, goal position q_f, and environment configuration O_j
Ensure: set of configurations Q_G, path length d, and flag $goal$
1: $APF \Leftarrow$ @ APF$(q_0, q_f, O_j, k_{rep}, k_{att})$
2: $t_0 \Leftarrow 0$
3: set N_g, m, and ℓ
4: initialize $P_0(t_0)$
5: initialize $\mathcal{S}_0$ and $\mathcal{S}_i$, $1 \leq i \leq m$
6: **while** $t_0 < N_g$ **do**
7: **for** each elementary membrane $\mathcal{S}_i$ **do**
8: $t_i \Leftarrow 0$
9: evaluate each bacterium of $P_i(t_i)$ in parallel using APF
10: **while** not termination **do**
11: copy the best bacteria $R \Leftarrow P_i(t_i)$
12: perform selection $P' \Leftarrow P_i(t_i)$
13: perform crossover P'
14: perform bacterial mutation P'
15: evaluate each bacterium of P' in parallel using APF
16: $P_i(t_i + 1) \Leftarrow P' \cup R$
17: $t_i \Leftarrow t_i + 1$
18: **end while**
19: **end for**
20: merge all the elementary membranes $\mathcal{S}_i$, into $\mathcal{S}_F$
21: apply communication rules and send out $[k_{rep}, k_{att}]_{opt}$
22: divide $\mathcal{S}_F$ in m elementary membranes $\mathcal{S}_i$
23: $t_0 \Leftarrow t_0 + 1$
24: **end while**
25: $[Q_G, d, goal] \Leftarrow APF$ using $[k_{rep}, k_{att}]_{opt}$

The objective of the membrane pseudo-bacterial potential field algorithm is to obtain a sequence of positions $Q_G = [q_0, q_1, \ldots, q_f]$, i.e., a collision-free path that will guide the mobile robot to reach the goal position with the minimum path length. Consequently, Algorithm 1 performs the path planning, with the characteristic attribute that offers an optimal or nearly optimal set of configurations Q_G. Other parameters that Algorithm 1 yields are the path length d, and a Boolean parameter $goal$ that shows if the goal position was reached.

In line 1 of Algorithm 1 a function handler APF (artificial potential field method) is employed to simplify the pseudocode writing. This APF function handler is equivalent to writing the APF pseudocode of Algorithm 2 with all the input parameters. In lines 2 and 3, a generation counter t_0 is set to zero and the parameters, number of generations N_g, number of elementary membranes m, and subpopulation size ℓ are established, respectively. In line 4, an initial random population $P_0 = \{P_1, P_2, \ldots, P_m\}$ of bacteria is generated. Where, P_i is a subpopulation, $P_i = \{p_1, \ldots, p_\ell\}$, $1 \leq i \leq m$, and $p_j = [k_{rep}, k_{att}]$ is a bacterium codified with the information of the proportional

gains k_{rep}, and k_{att}, i.e., a potential solution for the path planning problem presented. Then, the population will be divided into m subpopulations P_i that will be distributed between the elementary membranes $\mathcal{S}_i$.

In line 5, the Algorithm 1 initializes the membrane structure which is composed of an outer membrane (skin membrane) denoted by $\mathcal{S}_0$ and m number of inner membranes (elementary membranes) denoted by $\mathcal{S}_i$. In this regard, the membrane pseudo-bacterial potential field algorithm uses the hierarchical structure of a cell-like P system, specifically a one-level membrane structure with m number of elementary membranes contained in the skin membrane. Therefore, each elementary membrane $\mathcal{S}_i$ will be initialized with a subpopulation P_i composed of ℓ bacteria.

From lines 6 to 24, there is an iterative process of the membrane pseudo-bacterial potential field algorithm to find the global best bacterium $[k_{rep}, k_{att}]_{opt}$ through the evolution of the population. From lines 7 to 19, each elementary membrane $\mathcal{S}_i$ evolves its corresponding subpopulation of bacteria through a pseudo-bacterial genetic algorithm indicated from lines 10 to 18 of Algorithm 1. For the sequential implementation, the evaluation of each bacterium employs the artificial potential field (APF) function in sequential form, i.e., first, the bacterium p_1 is evaluated, after that, the second bacterium p_2 is evaluated and so on until the last bacterium p_ℓ is evaluated in the subpopulation P_i. On the other hand, for the parallel implementation, the evaluation of each bacterium is in a concurrent form regarding the number of available cores in the CPU or GPU as appropriate. Therefore, before the iterative process of the pseudo-bacterial genetic algorithm, in lines 9 and 15, each subpopulation P_i is evaluated using the APF function (Algorithm 2) to obtain the path length d, i.e., the fitness value of each bacterium.

3.2 *Artificial Potential Field Function*

Algorithm 2 presents the pseudocode for the artificial potential field function that acts as a fitness function and as a local planning method for the membrane pseudo-bacterial potential field algorithm. The artificial potential field function employs the input parameters: start position q_0, goal position q_f, environment information O_j, repulsive potential field proportional gain k_{rep}, and attractive potential field proportional gain k_{att}. The output parameters of the Algorithm 2 are the set of configurations Q_G (path), the cost d (path length), and the flag $goal$.

From lines 1 to 4 of Algorithm 2 some parameters of the artificial potential field function are initialized. The core component of the Algorithm 2 is the artificial potential field hard-computing mathematical method to perform the path planning employing the set of proportional gains given by the membrane pseudo-bacterial potential field algorithm. The path is generated through an iterative process described in lines 5 to 15. The parameters d_G, τ, and $safe$ are the stop conditions for the loop, as can be seen in line 5.

From lines 16 to 22 of Algorithm 2, a goal condition is validated. If the last position q_τ is equal to the goal position q_f the flag $goal$ takes a true state and the

Algorithm 2 Artificial potential field function

Require: start position q_0, goal position q_f, environment configuration O_j, proportional gains k_{rep} and k_{att}
Ensure: set of configurations Q_G, path length d, and flag $goal$
$\tau \Leftarrow 0$ ▷ τ is the number of configurations
$d \Leftarrow 0$
$safe \Leftarrow$ True
$d_G \Leftarrow \|q_f - (q_\tau \Leftarrow q_0)\|$
while $d_G > 0$ and $\tau < N$ and $safe$ **do** ▷ N is the maximum number of configs.
  compute $U(q_\tau)$ using Eq. (2)
  compute $F(q_\tau)$ using Eq. (5)
  $q_{\tau+1} \Leftarrow q_\tau + \eta * F(q_\tau)/ \parallel F(q_\tau) \parallel$ ▷ η is the step size
  $d_G \Leftarrow \|q_f - q_{\tau+1}\|$
  $d \Leftarrow d + \parallel q_{\tau+1} - q_\tau \parallel$
  if $\rho \leq r + r_o$ **then** ▷ r is the mobile robot's radius
    $safe \Leftarrow$ False
  end if
  $\tau \Leftarrow \tau + 1$
end while
if q_τ is equal to q_f **then**
  $goal \Leftarrow$ True
  $Q_G \Leftarrow [q_0, q_1, \ldots, q_\tau]$
else
  $goal \Leftarrow$ False
  $d \Leftarrow \infty$
end if

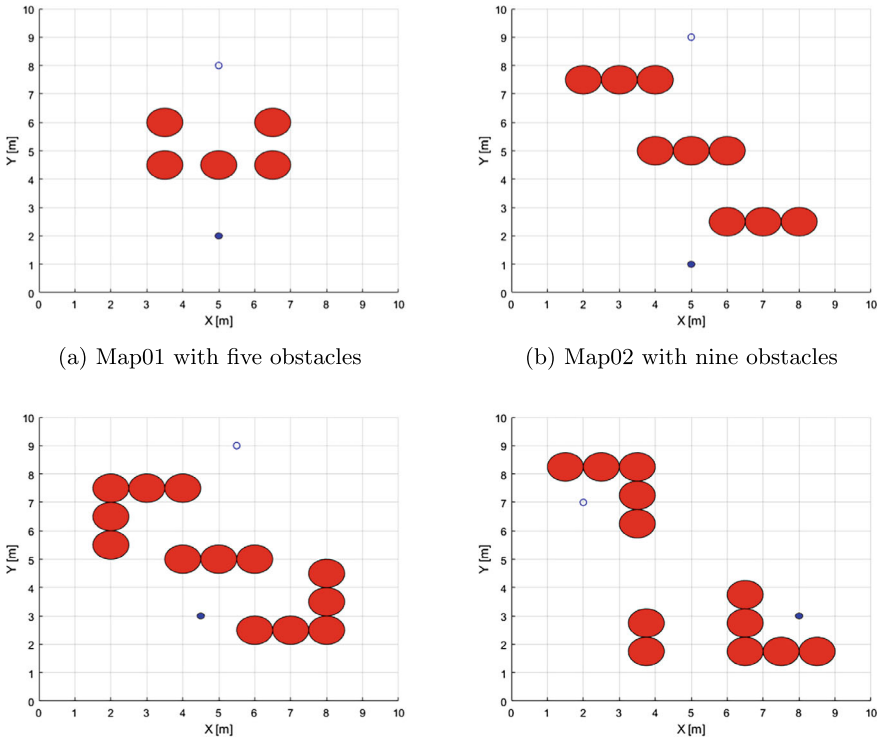
resultant path Q_G is composed by the set of configurations from q_0 to q_τ. If not, the flag $goal$ takes a false state, and the cost d takes an infinite value (very high value).

4 Experiments and Results

This section describes the experiments and presents the results for mobile robot path planning regarding path length for four test environments. Furthermore, we present a comparative analysis of three different implementations of the membrane pseudo-bacterial potential field algorithm to evaluate the acceleration of the path planning computation. The sequential implementation on CPU in C++, the parallel implementation on CPU in C++ using the open multi-processing (OpenMP) application programming interface, and the parallel implementation on GPU in C++ using CUDA (compute unified device architecture).

The experiments were conducted on an Intel Core i9 CPU (3.60 GHz) with 16 GB of RAM and an NVIDIA GeForce RTX 2080 Super GPU running Ubuntu 22.04 with CUDA 12.1.

The following setup parameters granted the best results for the three different implementations:



(a) Map01 with five obstacles

(b) Map02 with nine obstacles

(c) Map03 with thirteen obstacles

(d) Map04 with twelve obstacles

Fig. 4 Test environments

1. The population size was fixed to 16 bacteria and doubled in each test up to 512 bacteria to assess the membrane pseudo-bacterial potential field algorithm performance.
2. The bacteria were codified using two genes, k_{rep} and k_{att}.
3. The proportional gains were constrained with values greater than zero and less than or equal to ten.
4. A single-point crossover was used.
5. The mutation rate was fixed at 0.20.
6. The elitist selection was used.
7. The selection rate was fixed at 0.50.
8. The stop condition is the maximum number of generations; it was fixed to ten for the pseudo-bacterial genetic algorithm.
9. The membrane selection rate was fixed at 0.25.
10. The stop condition was fixed to ten for the skin membrane.

The test environments are shown in Fig. 4, where the origin or the (0.00, 0.00) coordinates are at the bottom left corner, and the dimensions of each test environment

are 10×10 m. The test environments show the start position, the goal position, and the localization of the obstacles. Each test environment holds a label for its identification as Map01, Map02, Map03, and Map04.

Figure 4a shows Map01, which contains five circular obstacles forming a U-shaped obstacle. In this test environment, the path planning problem is to go from the start position $q_0 = (5.00, 8.00)$ to the goal position $q_f = (5.00, 2.00)$. Figure 4b shows Map02, which contains nine circular obstacles forming three stepped blocks. In this test environment, the path planning problem is to go from the start position $q_0 = (5.00, 9.00)$ to the goal position $q_f = (5.00, 1.00)$. Figure 4c shows Map03, which contains thirteen obstacles forming two L-shaped obstacles and one horizontal obstacle with a rectangular form. In this map, the path planning problem is to go from the start position $q_0 = (5.00, 9.00)$ to the goal position $q_f = (4.50, 3.00)$. Figure 4d shows Map04, which contains twelve obstacles forming two L-shaped obstacles and one vertical obstacle with a rectangular form. The path planning problem in this map is to go from the start position $q_0 = (2.00, 7.00)$ to the goal position $q_f = (8.00, 3.00)$.

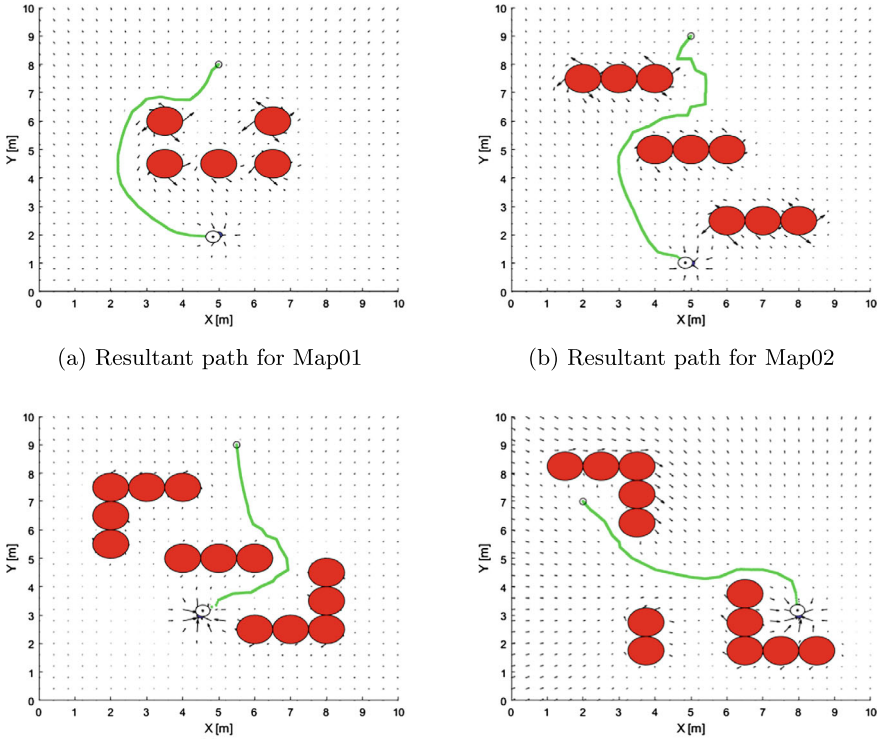
4.1 Path Planning Results

Figure 5 shows the best resultant path obtained with the membrane pseudo-bacterial potential field algorithm in each test environment and Fig. 6 shows its corresponding potential field surface.

Figure 5a shows the best resultant path for Map01, it has a path length of 9.304 m, and it was generated with the proportional gains $k_{rep} = 5.147$ and $k_{att} = 0.874$, Fig. 6a shows its resultant potential field surface. For Map02, Fig. 5b shows the best path obtained. The path has a length of 11.167 m, and it was generated with the proportional gains $k_{rep} = 4.968$ and $k_{att} = 0.699$, Fig. 6b shows its resultant potential field surface. Figure 5c shows the best resultant path for Map03, which has a length of 7.734 m, and it was generated with the proportional gains $k_{rep} = 0.947$ and $k_{att} = 0.352$, Fig. 6c shows its resultant potential field surface. The best resultant path for Map04 is shown in Fig. 5d, it has a path length of 8.283 m, and it was generated with the proportional gains $k_{rep} = 1.876$ and $k_{att} = 0.730$, Fig. 6d shows its resultant potential field surface.

Table 1 shows the results obtained with the membrane pseudo-bacterial potential field algorithm with different population sizes. The results show the proportional gains k_{rep} and k_{att} for the best resultant path obtained in terms of length. The results show the best, mean, worst, and standard deviation of ten independent tests in each population size. The results show that the membrane pseudo-bacterial potential field algorithm finds the shortest path when the population size is 512 bacteria. We can see that the quality of the solutions, i.e., the shortest path found, improves as the population grows.

Figure 7 shows the variability or dispersion of the path planning results for the membrane pseudo-bacterial potential field algorithm in terms of path length consid-



(a) Resultant path for Map01

(b) Resultant path for Map02

(c) Resultant path Map03

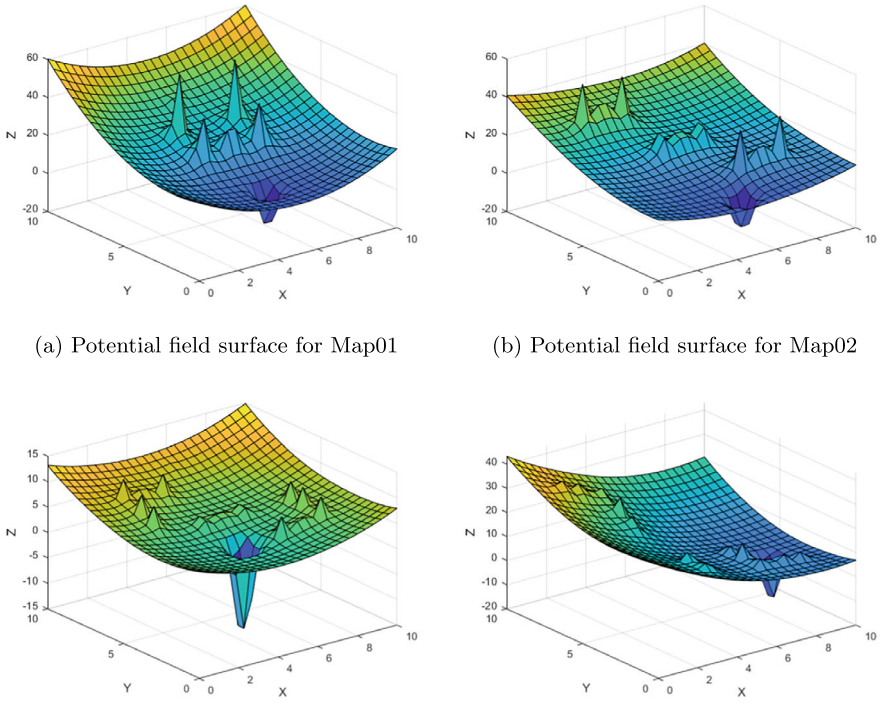
(d) Resultant path for Map04

Fig. 5 The best results for each test environment

ering six different population sizes. We can observe a comparison in the range and distribution of the path length results for the six population sizes employed. Furthermore, we can observe a more extended box that indicates more dispersed data. Additional information is presented by the overall spread, as shown by the extreme values at the end of the two whiskers. Also, in some cases, we can observe larger ranges, that indicate wider distribution, that is, more scattered data.

4.2 Performance Results

Table 2 shows the performance results of ten independent run tests to obtain the mean time of execution and the standard deviation (S.D.) for the membrane pseudo-bacterial potential field algorithm in each implementation, the sequential CPU, the parallel CPU, and the parallel GPU. The total of evaluated (T. Eval.) bacteria in each test is based on the product of the population size, the maximum number of generations in the pseudo-bacterial genetic algorithm, the number of elementary



(a) Potential field surface for Map01

(b) Potential field surface for Map02

(c) Potential field surface for Map03

(d) Potential field surface for Map04

Fig. 6 Resultant potential field surface for each test environment

membranes, and the maximum number of generations in the skin membrane, e.g., for a population size of 16, a maximum number of generations in the pseudo-bacterial genetic algorithm of 10, considering 2 elementary membranes, and a maximum number of generations in the skin membrane of 10, we have $16 \times 10 \times 2 \times 10 = 3,200$ evaluated bacteria.

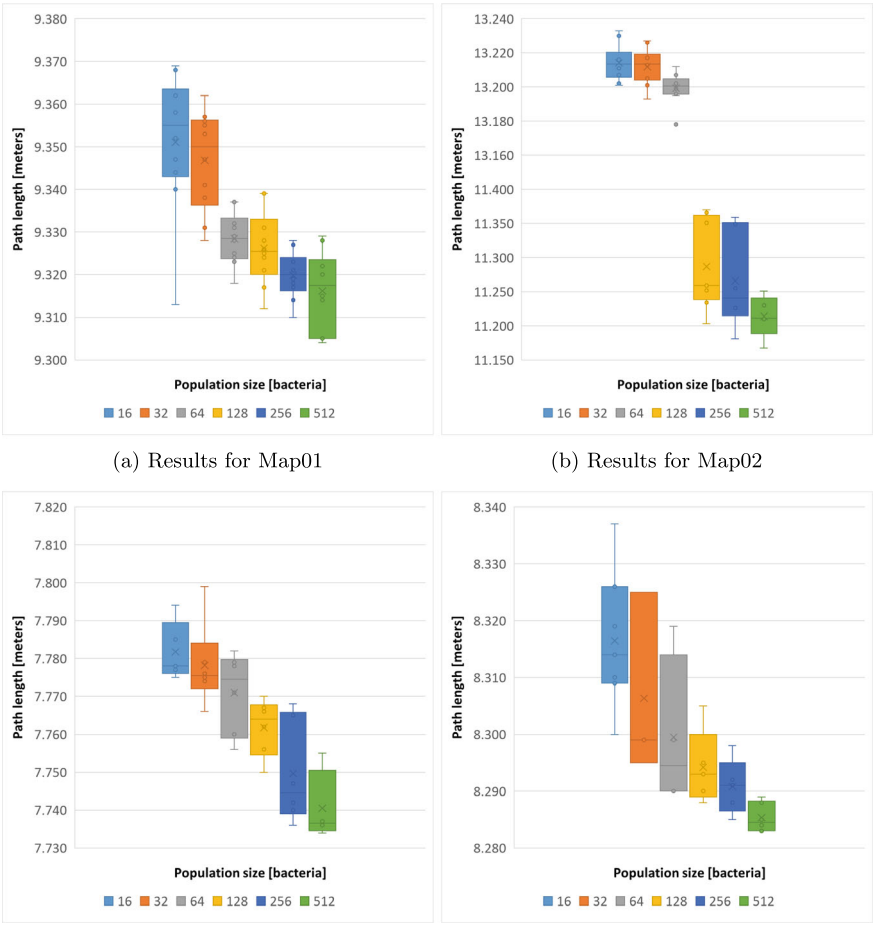
In Table 2, we can observe that the best speedup achieved for the parallel implementation in CPU is $8,669.51/1,129.48 = 7.68$, and for the parallel implementation in GPU is $8,669.51/86.30 = \mathbf{100.45}$ in Map01 for 102,400 evaluations. In Map02, the results show that the best speedup achieved for the parallel implementation in CPU is $3,547.78/454.07 = 7.81$ for 25,600 evaluations, and for the parallel implementation in GPU is $12,385.89/99.30 = \mathbf{124.73}$ for 102,400 evaluations. In Map03, the results show that the best speedup achieved for the parallel implementation in CPU is $12,532.41/1,549.93 = 8.09$, and for the parallel implementation in GPU is $12,532.41/93.80 = \mathbf{133.61}$ for 102,400 evaluations. In Map04, we observe that the best speedup for the parallel implementation in CPU is $12,032.08.20/1,556.70 = 7.73$ and for the parallel implementation in GPU is

Table 1 Path planning results obtained by the membrane pseudo-bacterial potential field algorithm with different population size

	Pop. size	(k_{rep}, k_{att})	Path length in meters			
			Best	Mean	Worst	S.D.
Map01	16	(4.644, 0.787)	9.313	9.351	9.369	0.0157
	32	(7.669, 1.303)	9.328	9.347	9.362	0.0111
	64	(7.698, 1.309)	9.318	9.328	9.337	0.0058
	128	(5.155, 0.875)	9.312	9.326	9.339	0.0082
	256	(5.277, 0.895)	9.310	9.320	9.328	0.0052
	512	(5.147, 0.874)	9.304	9.316	9.329	0.0088
Map02	16	(8.274, 1.118)	13.201	13.214	13.233	0.0102
	32	(3.246, 0.633)	13.193	13.212	13.227	0.0102
	64	(3.619, 0.705)	13.178	13.199	13.212	0.0087
	128	(9.402, 1.324)	11.203	11.287	11.370	0.0611
	256	(4.944, 0.694)	11.181	11.266	11.359	0.0659
	512	(4.968, 0.699)	11.167	11.214	11.251	0.0278
Map03	16	(8.375, 3.182)	7.775	7.782	7.794	0.0070
	32	(1.586, 0.588)	7.766	7.778	7.799	0.0101
	64	(1.611, 0.607)	7.756	7.771	7.782	0.0098
	128	(1.589, 0.595)	7.750	7.762	7.770	0.0069
	256	(1.014, 0.377)	7.736	7.750	7.768	0.0124
	512	(0.947, 0.352)	7.734	7.741	7.755	0.0084
Map04	16	(1.363, 0.528)	8.300	8.316	8.337	0.0113
	32	(1.474, 0.563)	8.295	8.306	8.325	0.0133
	64	(1.468, 0.564)	8.290	8.300	8.319	0.0118
	128	(2.160, 0.840)	8.288	8.294	8.305	0.0059
	256	(1.826, 0.710)	8.285	8.291	8.298	0.0044
	512	(1.876, 0.730)	8.283	8.285	8.289	0.0024

$12,032.08/102.86 = \mathbf{116.97}$ for 102,400 evaluations. We can observe a great acceleration achieved by the GPU.

The objective of the performance evaluation is to provide the speedup among the two different parallel implementations, parallel CPU and parallel GPU. Figure 8 shows the overall performance results. We can observe that the speedup is increased in both cases, for the parallel CPU implementation, and the parallel GPU implementation when the number of evaluations grows. These results are due to the increment of the total evaluations. We can observe the advantages of the parallel CPU implementation for a small number of evaluations, and the advantage is more significant for the GPU implementation with many bacteria (solutions) to evaluate. The speedup is computed by the relation of the execution time in a processor and the runtime on a certain number of processors. Therefore, the computation of the speedup is obtained considering the execution time taken by the sequential implementation of the mem-



(a) Results for Map01

(b) Results for Map02

(c) Results for Map03

(d) Results for Map04

Fig. 7 Variation in path planning results considering the population size

brane pseudo-bacterial potential field algorithm and the execution time taken by the parallel implementation in CPU and GPU, respectively.

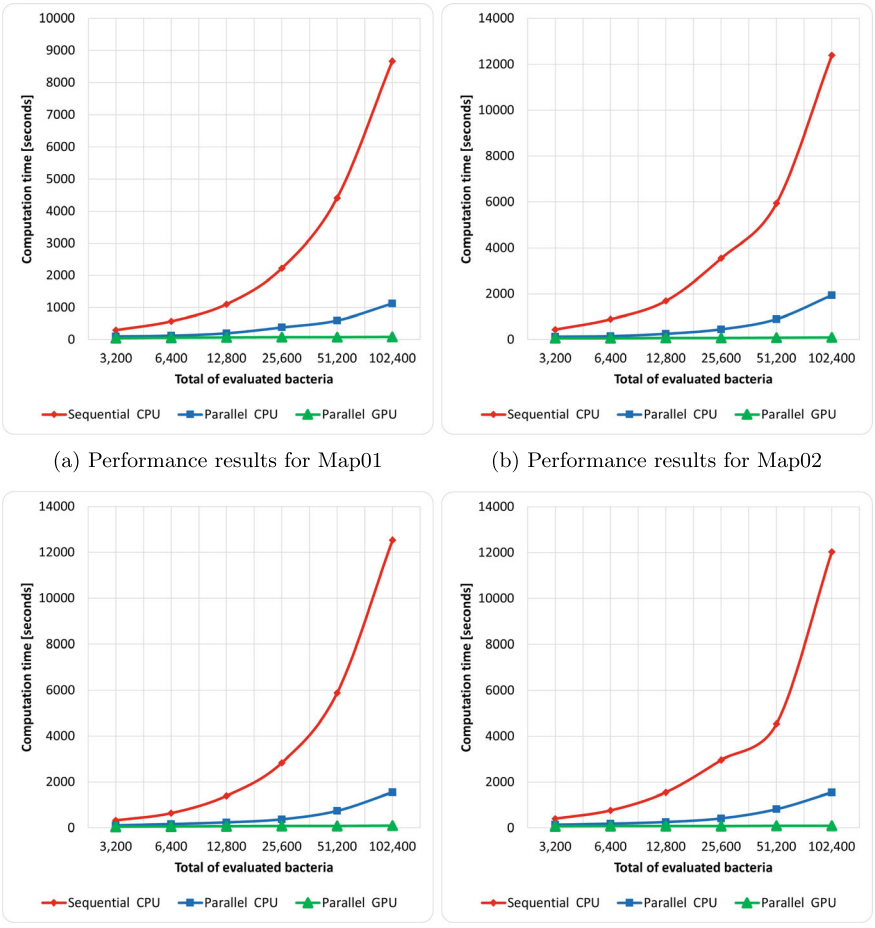
Table 2 Performance results obtained by the membrane pseudo-bacterial potential field algorithm with different number of evaluations

T. Eval.	Sequential CPU		Parallel CPU			Parallel GPU		
	Mean	S.D.	Mean	S.D.	Speedup	Mean	S.D.	Speedup
Map01								
3,200	298.71	32.08	103.75	12.69	**2.88**	47.42	1.65	**6.30**
6,400	574.66	41.30	125.70	10.33	**4.57**	58.28	2.55	**9.86**
12,800	1,103.17	50.96	200.54	11.95	**5.50**	68.69	1.71	**16.06**
25,600	2,220.42	85.69	383.34	50.80	**5.79**	75.07	1.54	**29.58**
51,200	4,409.02	191.86	597.61	11.07	**7.38**	81.58	1.78	**54.04**
102,400	8,669.51	177.23	1,129.48	35.95	**7.68**	86.30	1.90	**100.45**
Map02								
3,200	443.34	47.23	134.58	6.19	**3.29**	64.26	5.17	**6.90**
6,400	896.65	47.67	157.49	5.62	**5.69**	66.50	1.95	**13.48**
12,800	1,691.54	158.59	258.32	9.86	**6.55**	72.38	2.66	**23.37**
25,600	3,547.78	405.03	454.07	48.90	**7.81**	76.21	0.90	**46.55**
51,200	5,945.14	428.72	896.08	116.86	**6.63**	86.15	2.02	**69.01**
102,400	12,385.89	627.47	1,933.54	250.44	**6.41**	99.30	1.27	**124.73**
Map03								
3,200	322.58	29.82	107.95	8.68	**2.99**	58.22	6.48	**5.54**
6,400	639.63	69.44	164.70	20.73	**3.88**	67.82	5.65	**9.43**
12,800	1,395.52	238.44	239.04	19.00	**5.84**	75.20	2.77	**18.56**
25,600	2,838.82	518.48	371.58	17.17	**7.64**	83.61	0.92	**33.95**
51,200	5,883.06	824.91	742.26	19.09	**7.93**	84.35	0.45	**69.75**
102,400	12,532.41	1,178.10	1,549.93	60.07	**8.09**	93.80	1.76	**133.61**
Map04								
3,200	404.76	74.28	144.90	11.04	**2.79**	77.83	1.19	**5.20**
6,400	769.74	67.35	189.57	19.08	**4.06**	81.46	1.22	**9.45**
12,800	1,555.46	170.48	262.42	15.78	**5.93**	84.55	0.17	**18.40**
25,600	2,967.14	304.31	420.07	13.12	**7.06**	87.19	0.28	**34.03**
51,200	4,538.98	360.46	823.91	73.43	**5.51**	95.37	0.25	**47.60**
102,400	12,032.08	1,587.41	1,556.70	94.69	**7.73**	102.86	4.24	**116.97**

5 Conclusions

In this work, we have presented a membrane pseudo-bacterial potential field algorithm with GPU acceleration for mobile robot path planning. Across the simulation results, we can conclude that the membrane pseudo-bacterial potential field algorithm can be capable of facing complex and bigger path planning problems.

We observed that very small population sizes could guide the membrane pseudo-bacterial potential field algorithm to good path planning solutions, but the performance on the parallel implementations is not so good. In specific, for the GPU implementation, these results presented that the requirement for computationally



(a) Performance results for Map01

(b) Performance results for Map02

(c) Performance results for Map03

(d) Performance results for Map04

Fig. 8 Performance results

intensive is not met, i.e., the time spent on computation does not significantly exceed the time spent on transferring data to and from the GPU memory.

On the other hand, we observe that large population sizes could make the membrane pseudo-bacterial potential field algorithm spend more computation time in finding the solutions, with the benefit that these solutions are better in terms of path length. Also, for large population sizes, the benefits of parallel implementations are evident in terms of performance, for both, CPU implementation with a 8.09 of best speedup and GPU implementation with a 133.61 of best speedup.

Therefore, the GPU implementation can be powerful to accelerate because the path planning problem in this work has been stated as a data-parallel problem, and

we can observe the best performance results when we satisfy the computationally intensive and massively parallel criteria.

In conclusion, the performance results show that the parallel implementation on GPU accelerates the process by a factor above **133**× for the biggest population evaluated compared to sequential implementation on CPU. We can see the advantage of using the parallel implementation on GPU and that the membrane pseudo-bacterial potential field algorithm is highly scalable. Therefore, the proposed algorithm could be useful for many applications in mobile robots for global and local path planning, including industrial and domestic mobile robots, unmanned aerial vehicles, autonomous underwater vehicles, exploration vehicles, and self-driving cars.

References

1. Latombe, J.C.: Robot Motion Planning. Springer, New York, NY, USA (1991). https://doi.org/10.1007/978-1-4615-4022-9
2. Al Mahmud, S., Kamarulariffin, A., Ibrahim, A.M., Mohideen, A.J.H.: Advancements and challenges in mobile robot navigation: A comprehensive review of algorithms and potential for self-learning approaches. J. Intell. Robot. Syst. **110**(3) (2024). https://doi.org/10.1007/s10846-024-02149-5
3. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: Mobile robot path planning using membrane evolutionary artificial potential field. Applied Soft Computing 77, 236–251 (2019). https://doi.org/10.1016/j.asoc.2019.01.036
4. Orozco-Rosas, U., Picos, K., Montiel, O.: Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. IEEE Access 7, 156787–156803 (2019). https://doi.org/10.1109/ACCESS.2019.2949835
5. Orozco-Rosas, U., Picos, K., Montiel, O.: In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) Acceleration of Path Planning Computation Based on Evolutionary Artificial Potential Field for Non-static Environments. Studies in Computational Intelligence, vol. 862, pp. 271–297. Springer, Cham (2020). Chap. 22. https://doi.org/10.1007/978-3-030-35445-9_22
6. Hwu, W.-m.W., Kirk, D.B., Hajj, I.E.: Programming Massively Parallel Processors. Morgan Kaufmann, Cambridge, MA, USA (2022). https://doi.org/10.1016/C2020-0-02969-5
7. Păun, G.: Computing with membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000). https://doi.org/10.1006/jcss.1999.1693
8. Nawa, N.E., Furuhashi, T., Hashiyama, T., Uchikawa, Y.: A study on the discovery of relevant fuzzy rules using pseudobacterial genetic algorithm. IEEE Transactions on Industrial Electronics 46(6), 1080–1089 (1999). https://doi.org/10.1109/41.807990
9. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research 5(1), 90–98 (1986). https://doi.org/10.1177/027836498600500106
10. Zhang, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Verlan, S., Konur, S., Hinze, T., Gheorghe, M.: Membrane Computing Models: Implementations, 1st edn. Springer, Singapore, Singapore (2021). https://doi.org/10.1007/978-981-16-1566-5
11. Păun, G., Rozenberg, G.: A guide to membrane computing. Theoretical Computer Science 287(1), 73–100 (2002). https://doi.org/10.1016/S0304-3975(02)00136-6
12. Guo, P., Hou, M., Ye, L.: MEATSP: A membrane evolutionary algorithm for solving tsp. IEEE Access 8, 199081–199096 (2020). https://doi.org/10.1109/ACCESS.2020.3035058
13. Guo, P., Wang, X., Zeng, Y., Chen, H.: MEAMCP: A membrane evolutionary algorithm for solving maximum clique problem. IEEE Access 7, 108360–108370 (2019). https://doi.org/10.1109/ACCESS.2019.2933383

14. Guo, P., Quan, C., Chen, H.: MEAMVC: A membrane evolutionary algorithm for solving minimum vertex cover problem. IEEE Access 7, 60774–60784 (2019). https://doi.org/10.1109/ACCESS.2019.2915550
15. Liu, C., Shen, W., Zhang, L., Du, Y., Yuan, Z.: Spike neural network learning algorithm based on an evolutionary membrane algorithm. IEEE Access 9, 17071–17082 (2021). https://doi.org/10.1109/ACCESS.2021.3053280
16. Zhang, G., Cheng, J., Gheorghe, M., Meng, Q.: A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. Applied Soft Computing 13(3), 1528–1542 (2013). https://doi.org/10.1016/j.asoc.2012.05.032
17. Zhang, G., Cheng, J., Gheorghe, M.: Dynamic behavior analysis of membrane-inspired evolutionary algorithms. International Journal of Computers Communications & Control 9(2), 227–242 (2014). https://doi.org/10.15837/ijccc.2014.2.794
18. Zhang, G.-X., Gheorghe, M., Wu, C.-Z.: A quantum-inspired evolutionary algorithm based on p systems for knapsack problem. Fundamenta Informaticae 87(1), 93–116 (2008). https://doi.org/10.5555/2366091.2366098
19. Zhang, G., Gheorghe, M., Pan, L., Pérez-Jiménez, M.J.: Evolutionary membrane computing: A comprehensive survey and new results. Information Sciences 279, 528–551 (2014). https://doi.org/10.1016/j.ins.2014.04.007
20. Liu, C., Zhang, G., Liu, H., Gheorghe, M., Ipate, F.: An improved membrane algorithm for solving time-frequency atom decomposition. In: Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing, pp. 371–384. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11467-0_26
21. Nawa, N.E., Furuhashi, T.: Fuzzy system parameters discovery by bacterial evolutionary algorithm. IEEE Transactions on Fuzzy Systems 7(5), 608–616 (1999). https://doi.org/10.1109/91.797983
22. Picos, K., Orozco-Rosas, U.: Evolutionary correlation filtering based on pseudo-bacterial genetic algorithm for pose estimation of highly occluded targets. Multimedia Tools and Applications 80(15), 23051–23072 (2021). https://doi.org/10.1007/s11042-020-08991-7
23. Botzheim, J., Gál, L., Kóczy, L.T.: Fuzzy rule base model identification by bacterial memetic algorithms. In: Rakus-Andersson, E., Yager, R.R., Ichalkaranje, N., Jain, L.C. (eds.) Recent Advances in Decision Making, pp. 21–43. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02187-9_3
24. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation. International Journal of Advanced Robotic Systems 12(7), 81 (2015). https://doi.org/10.5772/60715
25. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) An Optimized GPU Implementation for a Path Planning Algorithm Based on Parallel Pseudo-bacterial Potential Field. Studies in Computational Intelligence, vol. 667, pp. 477–492. Springer, Cham (2017). Chap. 31. https://doi.org/10.1007/978-3-319-47054-2_31
26. Fogel, D.B.: An Introduction to Evolutionary Computation. Evolutionary Computation: The Fossil Record, p. 656. Wiley-IEEE Press, IEEE (1998). https://doi.org/10.1109/9780470544600.ch1
27. Zheng, X., Li, H., Zhang, Q., Liu, Y., Chen, X., Liu, H., Luo, T., Gao, J., Xia, L.: Intelligent decision-making method for vehicles in emergency conditions based on artificial potential fields and finite state machines. Journal of Intelligent and Connected Vehicles 7(1), 19–29 (2024). https://doi.org/10.26599/JICV.2023.9210025
28. Orozco-Rosas, U., Picos, K., Pantrigo, J.J., Montemayor, A.S., Cuesta-Infante, A.: Mobile robot path planning using a QAPF learning algorithm for known and unknown environments. IEEE Access 10, 84648–84663 (2022). https://doi.org/10.1109/ACCESS.2022.3197628
29. Yao, Q., Zheng, Z., Qi, L., Yuan, H., Guo, X., Zhao, M., Liu, Z., Yang, T.: Path planning method with improved artificial potential field—a reinforcement learning perspective. IEEE Access 8, 135513–135523 (2020). https://doi.org/10.1109/ACCESS.2020.3011211

30. Szczepanski, R., Tarczewski, T., Erwinski, K.: Energy efficient local path planning algorithm based on predictive artificial potential field. IEEE Access 10, 39729–39742 (2022). https://doi.org/10.1109/ACCESS.2022.3166632
31. Yuan, Z., Qin, H., Li, H.: An improved artificial potential field path planning based on gradient statistical mutation quantum genetic algorithm. IEEE Access 12, 119292–119301 (2024). https://doi.org/10.1109/ACCESS.2024.3450676
32. Tao, F., Ding, Z., Fu, Z., Li, M., Ji, B.: Efficient path planning for autonomous vehicles based on RRT* with variable probability strategy and artificial potential field approach. Sci. Rep. 14(1), 24698 (2024). https://doi.org/10.1038/s41598-024-76299-9
33. Orozco-Rosas, U., Picos, K., Montiel, O., Castillo, O.: In: Castillo, O., Melin, P. (eds.) Environment Recognition for Path Generation in Autonomous Mobile Robots. Studies in Computational Intelligence, vol. 827, pp. 273–288. Springer, Cham (2020). Chap. 19. https://doi.org/10.1007/978-3-030-34135-0_19
34. Peng, W., Zhang, Y., Guo, W.: Research on indoor UAV 3D path planning based on optimal ant colony-artificial potential field method. In: Tan, J., Liu, Y., Huang, H.-Z., Yu, J., Wang, Z. (eds.) Advances in Mechanical Design, pp. 795–813. Springer, Singapore (2024). https://doi.org/10.1007/978-981-97-0922-9_51
35. Luo, J., Wang, Z.-X., Pan, K.-L.: Reliable path planning algorithm based on improved artificial potential field method. IEEE Access 10, 108276–108284 (2022). https://doi.org/10.1109/ACCESS.2022.3212741
36. Zhai, L., Liu, C., Zhang, X., Wang, C.: Local trajectory planning for obstacle avoidance of unmanned tracked vehicles based on artificial potential field method. IEEE Access 12, 19665–19681 (2024). https://doi.org/10.1109/ACCESS.2024.3355952

Error Correction in Quantum Computing

Jesús Yaljá Montiel-Pérez, Jordi Fabián González-Contreras, Erik Zamora, and Juan Humberto Sossa-Azuela

Abstract This chapter presents a review of error correction in quantum computing, a crucial aspect for the development and practical implementation of quantum algorithms. It begins with a brief overview of classical error correction methods, highlighting their applications and the importance of ensuring data integrity in classical systems. Subsequently, the issue of noise in quantum systems is introduced, which significantly affects the fidelity of quantum operations due to the inherent fragility of quantum states. Modern quantum error correction techniques, such as Stabilizer Codes, the Shor Code, and the Steane Code, are explored, detailing how these methods can detect and correct errors in qubits. Specific examples of quantum computing are presented, illustrating the practical implementation of these techniques in quantum algorithms, as well as the current challenges and advancements in noise and error mitigation in quantum computing. This chapter provides a detailed and updated overview of the state-of-the-art in error correction and mitigation within the quantum domain, developed using the PRISMA methodology, emphasizing the importance of these techniques for the future of quantum computing.

J. Y. Montiel-Pérez (✉) · J. F. González-Contreras · E. Zamora · J. H. Sossa-Azuela
Centro de Investigación en Computación, Instituto Politécnico Nacional (CIC-IPN), Ciudad de México, Mexico
e-mail: jgonzalezc2023@cic.ipn.mx

E. Zamora
e-mail: ezamorag@ipn.mx

J. H. Sossa-Azuela
e-mail: hsossa@cic.ipn.mx

Avenida Juan de Dios Bátiz esquina Miguel Othón de Mendizábal Colonia Nueva Industrial, Ciudad de México, Mexico
e-mail: yalja@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_9

1 Introduction

In the modern era, the rapid acceleration of technological progress and the exponential growth of data have highlighted the need to develop a new vision of computing that surpasses the capabilities of classical systems. As we face increasingly complex problems in fields such as artificial intelligence, cryptography, material simulation, chemistry, and computational biology, the inherent limitations of traditional computers are becoming ever more apparent. Quantum computing emerges as a revolutionary solution, promising to transform how we approach and solve these challenges. By leveraging fundamental principles of quantum mechanics, such as superposition and entanglement, quantum computing can process information more efficiently than classical methods (in certain cases). This enables the execution of quantum algorithms that can solve specific problems exponentially faster while unlocking new possibilities for innovation in domains where traditional approaches have reached their limits.

The relevance of quantum computing to the advancement of modern computation lies in its ability to perform complex calculations at unimaginable speeds and its potential to revolutionize entire industries [1]. By investing in the research and development of quantum technologies, we can anticipate a new era of scientific discoveries, technological advancements, and practical applications that will redefine what is possible in the 21st century. This paradigm shift not only expands the boundaries of knowledge but also lays the foundation for more robust and advanced solutions to society's most pressing problems.

Research in quantum computing began in the 1980s, specifically in 1982, with Richard Feynman's seminal paper *"Simulating Physics with Computers"* [5]. This work was presented at the first Computational Physics Congress at MIT and published in volume 21 of the *International Journal of Theoretical Physics*. In his paper, Feynman discussed the feasibility and advantages of using quantum systems to simulate other quantum systems, thus laying the groundwork for the theory of quantum computation.

Since Feynman's time, quantum computing has made significant strides. Today, leading companies such as IBM, Google, Microsoft, and Honeywell are developing quantum hardware and algorithms that promise to revolutionize various domains of knowledge. For instance, in 2019, Google announced it had achieved quantum supremacy, with its quantum processor *Sycamore* completing a task in minutes that would have taken a classical supercomputer thousands of years. As better qubits are developed and error correction methods are refined, quantum computing is steadily moving closer to real-world practical applications for industry.

These advancements would not have been possible without the mathematical foundations that analytically prove these quantum phenomena. The contributions of mathematician David Hilbert are particularly important in the development of quantum computing. His work, especially the formalization of Hilbert space, laid the groundwork for the qubit, the basic unit of information in quantum computation. Without delving into too much detail, a Hilbert space is an algebraic concept that

provides a mathematical framework for describing and manipulating quantum states of qubits. Thanks to this mathematical structure, qubits can be represented not only in defined states, such as 0 or 1, but also in superpositions of these states. This allows qubits to perform parallel operations and handle information exponentially more efficiently than classical bits.

In the context of error correction and detection-fundamental to information theory and communication-these principles are particularly relevant. Their application ensures the integrity of data transmitted through noisy channels, where specially designed methods and codes for detecting and correcting errors play a crucial role, each with varying levels of complexity and effectiveness.

2 Error Detection and Correction

Error detection and correction is a fundamental aspect of information theory and communications. Its primary objective is to ensure the integrity of data transmitted through noisy channels, where errors may occur. Various methods and codes have been designed to detect and correct these errors [6].

2.1 Three-Bit Repeating Code

This is perhaps the simplest error correction code that exists, developed in the late 1970s in the paper "On the inherent intractability of certain coding problems" [37]. It essentially states that given an original information bit, $b \in \{0, 1\}$, the three-bit repetition code maps this bit into a sequence of three bits as follows:

$$\text{If } b = 0, \text{ then } 000.$$

$$\text{If } b = 1, \text{ then } 111.$$

More generally, for a bit string $x = x_1 x_2 \ldots x_n$, the three-bit repetition code maps each bit x_i into the corresponding sequence of three bits. For example, the message 101 would be encoded as 111000111.

2.2 Error Correction with the Three-Bit Repetition Code

The receiver can correct errors through *majority decoding*. Suppose we receive a sequence of three bits such as 110. The decoding is done by selecting the bit that appears most frequently. In this case, there are two 1's and one 0, so it would be decoded as 1.

This scheme can correct any single-bit error in a block of three bits, as a single-bit error will change one of the three bits, but the other two bits will match and prevail in the majority decoding.

2.3 Hamming Distance

The Hamming distance between two bit strings of equal length is the number of positions where the corresponding bits are different. Mathematically, the Hamming distance $d(x, y)$ between two strings x and y is defined as:

$$d(x, y) = \sum_{i=1}^{n} \mathbf{1}(x_i \neq y_i),$$

where $\mathbf{1}(x_i \neq y_i)$ is an indicator function that equals 1 if $x_i \neq y_i$ and 0 otherwise.

In the three-bit repetition code, the two possible encoded messages are 000 and 111. The Hamming distance between these two codes is 3, as all the bits differ. This minimum distance of 3 implies that the code can detect up to two errors and correct a single error in any block of three bits.

2.4 Relation to Hamming Distance

Since the Hamming distance of the code is 3, we can state that:

- **Error detection**: It can detect up to 2 errors (since the Hamming distance is 3, it can detect $d - 1 = 2$ errors).
- **Error correction**: It can correct up to 1 error (since error correction requires $\left\lfloor \frac{d-1}{2} \right\rfloor = 1$ correction).

Therefore, the three-bit repetition code is an example of a code that can detect two-bit errors and correct single-bit errors, illustrating the usefulness of Hamming distance in error correction theory. Below is a practical example.

Suppose we want to transmit the bit 1 using the three-bit repetition code. According to the encoding scheme, the bit 1 will be transmitted as:

Transmitted bit: 111.

Now, suppose that during transmission an error occurs in the second bit, and the receiver gets:

Received bits: 101.

To correct this error, the receiver uses the **majority decoding** method. In this case, the received bits are counted:

– There are two 1's and one 0.

Since the majority of the bits are 1, the receiver would apply a logical gate or invert the 0 to a 1, thus decoding the message as:

Decoded bit: 1.

This process corrects the single-bit error and ensures that the received message is correct.

3 Reed-Solomon Codes and Error Correction Techniques

Reed-Solomon codes, developed by Irving S. Reed and Gustave Solomon in 1960, represent an advanced error correction technology widely used in various applications, from storage on CDs and DVDs to space communications. These codes are particularly effective at correcting burst errors, allowing the correction of multiple errors in data blocks. In a contemporary context, as discussed in the article published at the 2021 International Symposium on Redundancy Issues in Information and Control Systems [7], Reed-Solomon codes are used as a foundation to improve the McEliece cryptosystem. This cryptographic system, known for its potential resistance to quantum computers, benefits from a significant reduction in the size of the public key by using Generalized Reed-Solomon (GRS) codes. This adaptation demonstrates how the fundamental principles of Reed-Solomon codes remain relevant and are innovatively applied in the realm of modern computer security.

A Reed-Solomon code is represented as RS(n, k), where n is the total number of symbols in the code block and k is the number of data symbols. The difference $n - k$ represents the number of parity symbols added to the data block, which allows the correction of up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ symbol errors [15].

3.1 Cyclic Redundancy Codes (CRC)

Cyclic Redundancy Codes (CRC) are a type of error detection code widely used in digital networks and storage devices. These codes are generated by calculating a hash of data through polynomial division by a predefined generator polynomial. Despite their redundant nature, CRCs are primarily used as error detection codes in standards such as IEEE 802.15.4 and Bluetooth Low Energy (BLE), due to the lack of a suitable decoder for use as error-correcting codes. Recent research explores methods to design decoders that enable CRCs to be used as error-correcting codes [8, 16, 17], thereby improving their utility in various applications, including IoT and low-power communications.

To calculate the CRC of a message $M(x)$ using a generator polynomial $G(x)$, the following operation is performed:

$$R(x) = M(x) \cdot x^r \mod G(x) \tag{1}$$

where r is the degree of the generator polynomial $G(x)$. The result $R(x)$ is appended to the original message, forming the encoded message. During reception, the integrity of the received message can be verified by recalculating the CRC and comparing it with the received CRC.

These error correction and detection methods have numerous practical applications. For example, the Hamming code is used in computer memory to correct single-bit errors. Reed-Solomon codes are crucial in error correction for optical disks and deep-space data transmission. CRCs are present in network communications, ensuring that data packets are not corrupted during transmission.

3.2 Parity Check Matrix and Generator Matrix for Error Correction

Two fundamental concepts in error correction coding theory are explored: the parity check matrix and the generator matrix. These matrices are key tools in the design and analysis of linear codes, which are used to detect and correct errors in data transmission.

3.2.1 Parity Check Matrix

The parity check matrix, commonly denoted as $\mathbf{H}$, is a matrix used to verify the parity of a received code. In a linear code (n, k), where n is the length of the code and k is the number of information bits, the matrix $\mathbf{H}$ has dimensions $(n - k) \times n$. This matrix enables efficient error detection by calculating syndromes from the received data, helping to identify erroneous bits within a code word.

The relationship between the code vector $\mathbf{c}$ and the matrix $\mathbf{H}$ is expressed as:

$$\mathbf{H} \cdot \mathbf{c}^T = \mathbf{0} \tag{2}$$

where $\mathbf{c}$ is a code vector of length n. If $\mathbf{c}$ satisfies this equation, it is considered a valid codeword of the linear code. Any deviation from this equality indicates the presence of errors [18, 19].

In the context of recent research on Applied Artificial Intelligence and Computing, Low-Density Parity-Check (LDPC) codes [9] are integrated, where LDPC codes are designed with matrices to check the code's parity, particularly using Progressive Edge Growth (PEG) techniques, which improve error correction capabilities

while minimizing average values. This LDPC-based approach not only enhances energy efficiency but also ensures robustness in wireless communication networks and multimedia transmissions, highlighting its versatility and effectiveness in modern telecommunications.

To construct $\mathbf{H}$, the generator matrix $\mathbf{G}$ of the code is used. The matrix $\mathbf{G}$ has dimensions $k \times n$ and its rows form a basis for the code space. The matrix $\mathbf{H}$ can be obtained as follows:

First, the matrix $\mathbf{G}$ is constructed in standard form: $\mathbf{G} = [\mathbf{I}_k|\mathbf{P}]$, where $\mathbf{I}_k$ is the identity matrix of size k and $\mathbf{P}$ is a matrix of size $k \times (n-k)$. Then, the matrix $\mathbf{H}$ is formed as $\mathbf{H} = [-\mathbf{P}^T|\mathbf{I}_{n-k}]$.

3.2.2 Generator Matrix

The generator matrix $\mathbf{G}$ is a matrix used to generate all possible codewords of a linear code. Given an information bit sequence $\mathbf{u}$ of length k, the corresponding codeword $\mathbf{c}$ of length n is obtained as:

$$\mathbf{c} = \mathbf{u} \cdot \mathbf{G} \tag{3}$$

For a linear code (7, 4), known as the Hamming (7,4) code, the generator matrix $\mathbf{G}$ for this code can be constructed as follows:

$$\mathbf{G} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{bmatrix} \tag{4}$$

There is a dual relationship between the generator matrix $\mathbf{G}$ and the parity check matrix $\mathbf{H}$. For a linear code, any code vector $\mathbf{c}$ must satisfy $\mathbf{H} \cdot \mathbf{c}^T = \mathbf{0}$, and it can be generated as $\mathbf{c} = \mathbf{u} \cdot \mathbf{G}$ for some information vector $\mathbf{u}$.

To verify if a received codeword $\mathbf{r}$ is valid, it is multiplied by $\mathbf{H}$:

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{r}^T \tag{5}$$

If $\mathbf{s} = \mathbf{0}$, then $\mathbf{r}$ is a valid codeword. If $\mathbf{s} \neq \mathbf{0}$, the vector $\mathbf{s}$, called the syndrome, indicates the presence and position of errors.

3.3 Classical Error Correction

Classical error correction in communication can be considered, without loss of generality, as the communication of binary digit strings, i.e., the binary vectors introduced in the previous section. A given binary vector, also called a binary word, that is to be

sent from point A to point B is called a message. A noisy communication channel may corrupt the message; however, since the message $\mathbf{u}$ is a binary vector, the only effect that noise can have is to change it to another binary vector $\mathbf{u}'$. The difference $\mathbf{e} = \mathbf{u}' - \mathbf{u}$ is called the error vector. Error correction consists of deducing the original message $\mathbf{u}$ from the received message $\mathbf{u}'$.

To address this problem, error correction codes are used, which add redundancy to the original message before transmission. This redundancy allows errors in the received message to be detected and corrected. A classic example is the Hamming code, which is capable of correcting single-bit errors and detecting two-bit errors. In this scheme, generator matrices and parity check matrices are used to encode and decode messages. The generator matrix $\mathbf{G}$ transforms the information message into a code, while the parity check matrix $\mathbf{H}$ allows verifying and correcting errors in the received message.

The key to error correction lies in the ability to identify the position of errors through the calculation of the syndrome, $\mathbf{s} = \mathbf{H} \cdot \mathbf{u}'^{T}$. If the syndrome is zero, the received message is valid; if not, the syndrome provides information about the location of the error. This process is essential in digital communication systems, where error correction ensures data integrity.

3.4 Minimum Distance Coding

The noisy channel described is known as a binary symmetric channel. This is a binary channel (the only type considered in this chapter) where noise independently affects each bit sent through the channel. Additionally, it is symmetric, meaning the channel causes $0 \rightarrow 1$ and $1 \rightarrow 0$ errors with equal probability [20].

When n bits are sent through a binary symmetric channel, the probability that m of them are flipped ($0 \leftrightarrow 1$) is the probability of m independent events occurring in n trials:

$$P(m \text{ bits flipped}) = C(n, m)p^m(1-p)^{n-m} \tag{6}$$

where the binomial coefficient is $C(n, m) = \frac{n!}{m!(n-m)!}$.

Minimum distance coding is a technique used to improve the reliability of data transmission over noisy channels. The minimum distance of a code, denoted as d_{min}, is the smallest Hamming distance between any pair of codewords. The Hamming distance between two codewords is the number of positions in which they differ. For a code to detect up to t errors, it is required that $d_{min} \geq t + 1$, and to correct up to t errors, $d_{min} \geq 2t + 1$ is needed.

For example, suppose $n = 7$ bits are sent through a binary symmetric channel with an error probability $p = 0.1$. The probability of exactly $m = 2$ bits being flipped can be calculated as follows:

$$P(m = 2) = C(7, 2) \cdot (0.1)^2 \cdot (0.9)^5 \tag{7}$$

The binomial coefficient $C(7, 2)$ is:

$$C(7, 2) = \frac{7!}{2!(7-2)!} = \frac{7 \cdot 6}{2 \cdot 1} = 21 \tag{8}$$

Thus, the probability of exactly 2 bits being flipped is:

$$P(m = 2) = 21 \cdot (0.1)^2 \cdot (0.9)^5 = 21 \cdot 0.01 \cdot 0.59049 = 0.12321 \tag{9}$$

Therefore, the probability that exactly 2 of the 7 bits transmitted are flipped in a binary symmetric channel with an error probability of 0.1 is approximately 0.12321.

This example illustrates how error probabilities can be calculated for a binary symmetric channel and highlights the importance of using error-correcting codes with an appropriate minimum distance to improve communication reliability. Minimum distance coding ensures that errors introduced by noise can be detected and corrected, maintaining the integrity of transmitted data.

3.5 Limits on Code Size

To communicate k bits of information using an error-correcting code, $n > k$ bits must be sent through the channel. The ratio k/n, called the code rate, provides a measure of the cost of error correction. Various bounds on k/n can be derived [21].

In a Hamming space of n dimensions [22], consisting of n-bit vectors, there are $C(n, t)$ error vectors of weight t. Each member of a code correcting t errors has $\sum_{i=0}^{t} C(n, i)$ erroneous versions (including the error-free version) that must be distinct from other codewords and their erroneous versions to enable error correction. These versions are also distinct among themselves because $e_1 \neq e_2 \Rightarrow \mathbf{u} + e_1 \neq \mathbf{u} + e_2$.

However, there are only 2^n possible vectors in the entire Hamming space. Thus, the number of codewords m in error-correcting codes of n bits is constrained by the Hamming bound:

$$m \sum_{i=0}^{t} C(n, i) \leq 2^n \tag{10}$$

This is also called the sphere-packing bound, as the problem is equivalent to packing as many spheres as possible in an n-dimensional space, where each sphere has a radius t. For linear codes, $m = 2^k$, so the Hamming bound becomes:

$$2^k \sum_{i=0}^{t} C(n, i) \leq 2^n \tag{11}$$

For instance, with $n = 7$ and $t = 1$, the number of maximum codewords m in a code that corrects a single error is determined as follows:

$$\sum_{i=0}^{1} C(7, i) = C(7, 0) + C(7, 1) = 1 + 7 = 8 \tag{12}$$

Applying the Hamming bound:

$$m \cdot 8 \leq 2^7 \Rightarrow m \leq \frac{2^7}{8} = \frac{128}{8} = 16 \tag{13}$$

The maximum number of codewords m in a 7-bit code that can correct a single error is 16. This means that, with a single error in any transmitted 7-bit codeword, the receiver can identify and correct the erroneous bit for codes with a maximum of 16 codewords.

This example demonstrates how the Hamming bound imposes restrictions on the size of error-correcting codes. The code rate k/n and the error correction capability are directly influenced by these bounds. In practice, error-correcting code designers must balance the code rate and error correction capability to optimize communication reliability over noisy channels.

3.6 Linear Codes and Error Syndrome

The importance of linear codes lies in their convenience, particularly in the speed with which any received erroneous word can be corrected. They are also significant in generalizing classical error correction to quantum error correction [24].

So far, the only error correction method discussed is the straightforward idea of a lookup table, where a received vector $\mathbf{w}$ is compared to all codewords $\mathbf{u} \in C$ and their erroneous versions $\mathbf{u} + \mathbf{e}$, $\mathbf{e} \in S$, until a match $\mathbf{w} = \mathbf{u} + \mathbf{e}$ is found, and the vector is corrected to $\mathbf{u}$. This method is inefficient in terms of time or memory resources, as there are 2^n vectors $\mathbf{u} + \mathbf{e}$ [23].

For linear codes, a significant improvement is achieved by calculating the error syndrome $\mathbf{s}$ given by:

$$\mathbf{s} = \mathbf{H}\mathbf{w}^T \tag{14}$$

where $\mathbf{H}$ is the parity-check matrix. The error syndrome provides information about the error location without the need to compare each received vector to a lookup table.

For example, consider the Hamming (7, 4) code, which can correct single-bit errors. The parity-check matrix $\mathbf{H}$ for this code is:

$$\mathbf{H} = \begin{bmatrix} 1\,1\,0\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,1\,0\,0\,1 \end{bmatrix} \tag{15}$$

If the codeword $\mathbf{c} = [1, 0, 1, 1, 0, 1, 0]$ is transmitted, but due to noise, the received vector is $\mathbf{w} = [1, 0, 1, 1, 1, 1, 0]$, the syndrome $\mathbf{s}$ is:

$$\mathbf{s} = \mathbf{H}\mathbf{w}^T = \begin{bmatrix} 1\,1\,0\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,1\,0\,0\,1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

The syndrome $\mathbf{s} = [1, 0, 0]^T$ indicates that the error is located in the 5th position. Correcting the error yields the original codeword $\mathbf{c}$.

Linear codes and error syndromes allow for efficient error detection and correction, making them essential for classical and quantum error correction.

4 Quantum Error Correction

In the previous section, a general introduction to classical error detection and correction was presented. This theory is fundamental for error correction in quantum computing since the principles used are very similar.

Quantum error correction (QEC) relies on three central ideas: noise digitization, manipulation of error operators and syndromes, and the construction of quantum error correction codes (QECC). The degree of success of QEC depends on the physics of noise; this is addressed after discussing the three central ideas.

However, analogous to the Three-Bit Repetition Code, there is also a quantum equivalent, the Three-Qubit Quantum Repetition Code.

4.1 Three-Qubit Quantum Repetition Code

The **Three-Qubit Quantum Repetition Code** is a fundamental method in quantum error correction theory that protects quantum states from common errors, such as *bit-flip* and *phase-flip*. This code uses three qubits to encode a single logical qubit.

4.2 Encoded Logical States

To protect a quantum qubit, the logical states $|0\rangle_L$ and $|1\rangle_L$ are encoded using three qubits as follows:

$$|0\rangle_L = |000\rangle, \quad |1\rangle_L = |111\rangle.$$

This means that the logical state $|0\rangle_L$ is a superposition of the three qubits in the state $|0\rangle$, and the logical state $|1\rangle_L$ is a superposition of the three qubits in the state $|1\rangle$.

4.3 Common Errors: Bit-Flip and Phase-Flip

In a quantum system, two of the most common errors that can occur are:

- **Bit-Flip** (X): This error is analogous to a classical bit flip, i.e., it converts $|0\rangle$ to $|1\rangle$ and vice versa. For example, applying an X operation to the first qubit in $|000\rangle$ changes it to $|100\rangle$.
- **Phase-Flip** (Z): This error introduces a phase change in the quantum state, converting $|+\rangle$ to $|-\rangle$ and vice versa, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. For example, applying a Z operation to the state $|+\rangle$ results in $|-\rangle$.

4.4 Superposition States

The quantum repetition code not only protects the basis states $|0\rangle$ and $|1\rangle$ but also superposition states. For example, a general superposition state of a qubit can be written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Using the three-qubit repetition code, this state is encoded as:

$$|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle.$$

This encoded state is robust against both bit-flip and phase-flip errors on a single qubit.

4.5 Bit-Flip Detection and Correction

Suppose a bit-flip error (X) occurs on the second qubit of the encoded state $|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$. This error changes the state to:

$$|\psi_{\text{error}}\rangle = \alpha|010\rangle + \beta|101\rangle.$$

To detect the error, we perform syndrome measurements using the stabilizer operators:

$$Z_1Z_2, \quad Z_2Z_3.$$

where Z_i are Pauli-Z operators applied to qubit i. Measuring these operators gives the results:

$$Z_1Z_2 = -1, \quad Z_2Z_3 = +1.$$

This indicates that the error has occurred on the second qubit. To correct the error, we apply the inverse operation X_2 to the second qubit, restoring the original state:

$$|\psi_{\text{corrected}}\rangle = \alpha|000\rangle + \beta|111\rangle.$$

4.6 Noise Digitization

Noise digitization involves transforming the continuous noise affecting qubits into a discrete set of errors that can be corrected. In the quantum case, errors are described through error operators acting on the qubits' states.

This method is based on the observation that any interaction between a set of qubits and another system (such as the environment) can be expressed in the form:

$$|\phi\rangle|\psi_0\rangle_e \to \sum_i (E_i|\phi\rangle)|\psi_i\rangle_e, \tag{17}$$

where each error operator E_i is a tensor product of Pauli operators acting on the qubits, $|\phi\rangle$ is the initial state of the qubits, and $|\psi_i\rangle_e$ are states of the environment, not necessarily orthogonal or normalized. Thus, general noise and/or decoherence can be expressed in terms of the Pauli operators σ_x, σ_y, and σ_z acting on the qubits. These operators are denoted as $X \equiv \sigma_x$, $Z \equiv \sigma_z$, and $Y \equiv -i\sigma_y = XZ$:

$$X \equiv \sigma_x, \quad Z \equiv \sigma_z, \quad Y \equiv -i\sigma_y = XZ. \tag{18}$$

To write tensor products of Pauli matrices acting on n qubits, we introduce the notation $X^u Z^v$, where u and v are n-bit binary vectors. Non-zero coordinates in u and v indicate where X and Z operators appear in the product. For example:

$$X \otimes I \otimes Z \otimes Y \otimes X \equiv X^{10011} Z^{00110}. \tag{19}$$

Error correction is the process of transforming a state like $E_i|\phi\rangle$ back to $|\phi\rangle$. X-error correction transforms $X^u Z^v|\phi\rangle$ into $Z^v|\phi\rangle$; Z-error correction transforms $X^u Z^v|\phi\rangle$ into $X^u|\phi\rangle$. Combining these, we arrive at the highly significant fact that to correct the most general noise possible (Eq. (21)), it is sufficient to correct only X and Z errors [25, 26].

Consider an example where a qubit is affected by bit-flip and phase-flip noise. Suppose the initial state of the qubit is $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$. Noise can transform this state into a combination of errors:

$$|\phi\rangle \to \sum_i E_i|\phi\rangle = a_0 I|\phi\rangle + a_1 X|\phi\rangle + a_2 Z|\phi\rangle + a_3 Y|\phi\rangle, \tag{20}$$

where a_0, a_1, a_2, a_3 are complex coefficients representing the probabilities of each type of error. If, for example, a bit-flip error (X) occurs, the state becomes:

$$X|\phi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle. \tag{21}$$

To correct this error, we apply the X operator:

$$X(\alpha|1\rangle + \beta|0\rangle) = \alpha|0\rangle + \beta|1\rangle = |\phi\rangle. \tag{22}$$

Similarly, if a phase-flip error (Z) occurs, the state becomes:

$$Z|\phi\rangle = Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle. \tag{23}$$

To correct this error, we apply the Z operator:

$$Z(\alpha|0\rangle - \beta|1\rangle) = \alpha|0\rangle + \beta|1\rangle = |\phi\rangle. \tag{24}$$

4.7 Construction of Quantum Error-Correcting Codes (QECC)

The construction of QECC involves designing quantum codes capable of detecting and correcting specific errors. These codes leverage principles of quantum superposition and entanglement to safeguard quantum information.

An example of a quantum error-correcting code is the Shor code, one of the earliest QECC developed. The Shor code protects one logical qubit using 9 physical qubits and can correct any single-qubit error (bit-flip, phase-flip, or both).

To protect the state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ using the Shor code, the original state is encoded into 9 qubits through the following steps [27]:

1. Repeat the initial state three times: $\alpha\,|000\rangle + \beta\,|111\rangle$.
2. Apply a three-qubit repetition code to each of the three qubits:

$$\alpha\,|000\rangle + \beta\,|111\rangle \rightarrow \alpha\,|000000000\rangle + \beta\,|111111111\rangle\,.$$

To detect and correct errors, parity syndromes are measured on blocks of three qubits. If an error is detected in any block, it can be corrected using the appropriate quantum operations [28].

4.8 Error Operators, Stabilizer, and Syndrome Extraction

We now examine the mathematics of error operators and syndromes, using the insightful approach proposed by Gottesman and Calderbank [12]. This introduced the concept of the stabilizer, which provided key insights in early discoveries and allowed significant generalizations to be written simply, remaining a cornerstone of recent research [10, 11].

Consider the set $\{I, X, Y, Z\}$ consisting of the identity and the three Pauli operators. The Pauli operators satisfy $X^2 = Y^2 = Z^2 = I$ and have eigenvalues ± 1. Two members of the set either commute ($XI = IX$) or anticommute ($XZ = -ZX$). Tensor products of Pauli operators, i.e., error operators, also square to one and either commute or anticommute. The term "error operator" here simply refers to "a product of Pauli operators"; such operators represent errors and sometimes parity checks in classical coding theory.

If there are n qubits in the quantum system, error operators have length n. The weight of an error operator is the number of terms not equal to I. For example, $X^{10011}Z^{00110}$ has a length of 5 and weight 4.

Let $H = \{M\}$ be a set of commuting error operators. Since these operators commute, they can have simultaneous eigenstates. Let $C = \{|u_i\rangle\}$ be an orthonormal set of simultaneous eigenstates with eigenvalue $+1$:

$$M|u_i\rangle = |u_i\rangle \quad \forall u \in C, \forall M \in H. \tag{23}$$

The set C forms a quantum error-correcting code, and H is its stabilizer. The orthonormal states $|u_i\rangle$ are called codewords or quantum codewords. In what follows, we focus on cases where H is a group. Its size is 2^{n-k} and is generated by $n - k$ linearly independent members of H. In this case, C has 2^k members, encoding k qubits since its members span a 2^k-dimensional subspace of the 2^n-dimensional Hilbert space of the

entire system. A general state in this subspace, called a coded or logical state, can be expressed as a superposition of the codewords:

$$|\phi_i^L\rangle = \sum_{u \in C} a_u |u_i\rangle. \tag{24}$$

A given QECC does not allow correction of all possible errors. Each code corrects a specific set $S = \{E\}$ of correctable errors.

The error operators in the stabilizer are all correctable, $E \in S \quad \forall E \in H$, since these operators have no effect on a general logical state (Eq. 24). If these operators are the only terms in the system's noise, the QECC is a noise-free subspace, also known as a decoherence-free subspace [28].

There is a larger set of errors that change the coded states but are still correctable by extracting an error syndrome and acting on the system based on the syndrome. It is shown that S can be any set of errors $\{E_i\}$ such that each product $E_1 E_2$ of two members belongs to H or anticommutes with a member of H. To see this, consider the latter case first:

$$E_1 E_2 M = -M E_1 E_2 \quad \text{for some} \quad M \in H. \tag{25}$$

The combined error operator $E_1 E_2$ is detectable. This is possible only if:

$$\{ME_1 = -E_1 M, ME_2 = E_2 M\} \quad \text{or} \quad \{ME_1 = E_1 M, ME_2 = -E_2 M\}. \tag{26}$$

To extract the syndrome, all observables in the stabilizer are measured. Measuring any $n - k$ linearly independent observables $M \in H$ suffices. Such measurements do not affect a state in the coded subspace, as such states are already eigenvectors of these observables. Measurement projects a noisy state onto an eigenvector of each M, with eigenvalues ± 1. The sequence of $n - k$ eigenvalues constitutes the syndrome. Equation (26) ensure that E_1 and E_2 have different syndromes, distinguishing them. When observable M is measured on the corrupted state $E|\phi_i^L\rangle$, (26) ensures a different eigenvalue is obtained for $E = E_1$ versus $E = E_2$. Thus, the error can be inferred from the syndrome and reversed by reapplying the inferred error operator to the system [29].

4.9 Conditions for Quantum Error Correcting Codes

The stabilizer-based discussion is useful because it focuses attention on the operators rather than the states. However, quantum codewords are very interesting states with a lot of symmetry and forms of entanglement. Quantum codewords in QECC can easily be shown to allow correction of the set S if and only if

$$\langle u_i | E_1 E_2 | v_i \rangle = 0 \tag{30}$$

$$\langle u_i|E_1E_2|u_i\rangle = \langle v_i|E_1E_2|v_i\rangle \quad (31)$$

for all E_1, $E_2 \in S$ and $|u_i\rangle$, $|v_i\rangle \in C$, with $|u_i\rangle \neq |v_i\rangle$. These are the requirements for quantum codewords for classical codes. In the case where E_1E_2 always anticommutes with a member of the stabilizer, we have that

$$\langle u_i|E_1E_2|u_i\rangle = \langle u_i|E_1E_2M|u_i\rangle = -\langle u_i|ME_1E_2|u_i\rangle = -\langle u_i|E_1E_2|u_i\rangle,$$

thus $\langle u_i|E_1E_2|u_i\rangle = 0$. This is a non-degenerate code; all code vectors and their erroneous versions are mutually orthogonal, and the quantum Hamming bound must be satisfied (see the next section).

The first condition states that, to be correctable, an error acting on a codeword must not produce a state that overlaps with another codeword or with an erroneous version of another codeword. This is what we would intuitively expect. Proof. Let R be the unitary operator describing the entire recovery operation (e.g., syndrome extraction followed by correction), then

$$R|\alpha_j\rangle E_j|v_i\rangle = |\alpha'_j\rangle|v_i\rangle \quad \Rightarrow \quad R|\alpha_i\rangle E_i|u_i\rangle = |\alpha'_i\rangle|u_i\rangle \quad \Rightarrow \quad \langle u_i|E_i^\dagger E_j|v_i\rangle = 0.$$

where $|\alpha_i\rangle$ are ancilla states, the environment, and any other system used during recovery. The last step uses the fact that the recovery must work when $|\alpha_i\rangle = |\alpha_j\rangle$ (as well as when $|\alpha_i\rangle \neq |\alpha_j\rangle$).

4.10 Code Construction

The main aspect of QECC stems from the physical ideas and mathematical techniques already discussed, combined with the fact that useful QECCs can actually be found. Code construction is itself a subtle and interesting area, which we will simply introduce here. First, we present an overview of stabilizer codes; this will connect the mathematics of error operators with that of binary vector spaces, which have been studied in classical coding theory. Examples will be examined in more detail in subsequent sections [30].

Let us first recall that stabilizer members are required to commute. It is easy to show that $X_uZ_v = (-1)^{u\cdot v}Z_vX_u$, where $u \cdot v$ is the binary parity operation, or dot product between binary vectors, evaluated in GF. From this, $M = X_uZ_v$ and $M' = X_{u'}Z_{v'}$ commute if and only if

$$u \cdot v' + v \cdot u' = 0.$$

The stabilizer is fully specified by writing any set of $n - k$ linearly independent error operators that span it. It is convenient to write these error operators as binary

strings u and v that indicate the X- and Z-parts, in the form of two binary matrices H_x, H_z of dimensions $(n-k) \times n$. The entire stabilizer is then uniquely specified by the binary matrix $(n-k) \times 2n$

$$H = (H_x|H_z),$$

and the requirement that all operators commute (i.e., H is an abelian group) is expressed by

$$H_x H_z^T + H_z H_x^T = 0,$$

where T denotes matrix transposition. The matrix H is the analogue of the parity-check matrix for a classical error-correcting code. The analogue of the generator matrix is the matrix $G = (G_x|G_z)$, which satisfies

$$H_x G_z^T + H_z G_x^T = 0.$$

In other words, H and G are duals with respect to the scalar product defined by $u \cdot v$. G has $n+k$ rows. H can be directly obtained from G by swapping the X- and Z-parts and extracting the usual binary dual of the resulting binary matrix $(n+k) \times 2n$.

This implies that G contains H. If G is the set of error operators generated by G, then G also contains H.

4.11 Steane's 7-Qubit Code

The simplest CSS code is the 7-bit code discovered by Steane [13]. It uses the CSS construction (38) with $H_1 = H_2 = H$. This is the Hamming code [7, 4, 3], corrects a single error, and contains its dual, leading to a single-error quantum error-correcting code with parameters $[[n, k, d]] = [[7, 1, 3]]$, i.e., a single qubit stored in 7 with minimum distance 3. The two code vectors are

$$|0\rangle_L = |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle, \quad (41)$$

$$|1\rangle_L = X|0\rangle_L.$$

More efficient QECCs than CSS codes exist. Good codes can be found by extending CSS codes and other methods. The perfect code $[[n, k, d]] = [[5, 1, 3]]$ encodes a single qubit ($k = 1$) and corrects all weight-1 errors (since $\frac{d-1}{1} = 1$) [31].

4.12 Shor's 9-Qubit Code

Shor's 9-qubit code [14] is a prominent example of a quantum error-correcting code that surpasses CSS codes in terms of efficiency. It uses the extended stabilizer code construction (CSS) to encode a single qubit into 9 qubits with a minimum distance of 3. This code has parameters [[9, 1, 3]], meaning it can correct up to one qubit error.

The two codewords are:

$$|0\rangle_L = |000000000\rangle + |111111111\rangle,$$

$$|1\rangle_L = X111111111|0\rangle_L.$$

This code is constructed using stabilizer and generator matrices. The stabilizer H and generator G are defined as:

$$H = \begin{pmatrix} 110000000 \\ 011000000 \\ 001100000 \\ 000011000 \\ 000000110 \end{pmatrix}, \quad G = \begin{pmatrix} H_x & H_z \\ 111111111 & 000000000 \\ 000000000 & 111111111 \end{pmatrix}.$$

Here, H_x and H_z represent the X and Z components of the error operators that form the stabilizer. This code can detect and correct a single qubit error due to its minimum distance of 3, making it more robust than codes with smaller distances.

4.13 Error Mitigation Techniques in Quantum Computers

Error mitigation is critical for the reliable operation of quantum computers due to the sensitivity of qubits to environmental perturbations and inherent errors. This section explores various techniques and notable examples of error mitigation in the context of quantum computers.

A fundamental technique for mitigating quantum errors is block encoding. Similar to classical coding theory, quantum error-correcting codes (QECC) are used here to protect quantum information from noise and hardware imperfections. An example is stabilizer codes like the CSS codes (Calderbank-Shor-Steane), which can detect and correct errors by measuring stabilizers and applying syndrome-based corrections.

The CSS code is a type of quantum error-correcting code that combines Pauli X and Z operators to protect quantum information. Consider the 7-qubit Hamming code, a well-known example of a CSS code.

This code has parameters [[7, 1, 3]], meaning it encodes a single qubit into 7 qubits with a minimum distance of 3. The two codewords are:

$$|0\rangle_L = |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle,$$

$$|1\rangle_L = X|0\rangle_L.$$

The stabilizer and generator matrices of the code can be defined as:

$$H = \begin{pmatrix} 1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0 \end{pmatrix}, \quad G = \begin{pmatrix} H_x & H_z \\ 1111111 & 0000000 \end{pmatrix}.$$

Here, H_x and H_z denote the X and Z parts of the error operators forming the stabilizer. This code detects and corrects errors using its stabilizer structure, allowing syndrome measurements to identify and apply appropriate corrections.

4.14 Dynamic Error Mitigation

Dynamic error mitigation is an active strategy that continuously adjusts quantum operations to compensate for detected errors. Techniques like quantum feedback monitor qubit states during execution and apply real-time adaptive corrections. This involves repeating operations or adjusting control parameters to maintain quantum coherence and reduce cumulative errors [32].

Quantum feedback is an advanced technique in quantum computing to monitor and correct errors in real time during quantum operations. Below is an example of implementing quantum feedback to improve qubit fidelity in a quantum system [33].

Suppose a quantum system performs a series of quantum gate operations $U(t)$ on a qubit. During the execution of $U(t)$, the qubit state is continuously monitored using non-destructive quantum measurement techniques such as partial state tomography.

If the qubit state deviates from the desired state due to quantum errors, a feedback mechanism is triggered. This mechanism may involve real-time adjustments to the control parameters of quantum gates, such as the amplitude and duration of microwave or laser pulses used to manipulate the qubits.

For illustration, consider a qubit intended to remain in the $|0\rangle$ state. During the execution of quantum operations, a Z-axis projection measurement determines the probability of finding the qubit in the $|1\rangle$ state. If this probability exceeds a predefined threshold, a correction pulse is applied to revert the qubit to the $|0\rangle$ state.

Quantum feedback corrects both systematic and stochastic errors affecting quantum operation accuracy. Integrating these techniques into dynamic quantum system control enhances qubit coherence and fidelity, essential for practical applications like quantum algorithms and simulations.

Developing and implementing effective quantum feedback techniques remain active research areas in quantum computing, aiming to build more robust and scalable systems for future technological applications.

4.15 Fault-Tolerant Codes

Fault-tolerant codes are essential for constructing scalable and robust quantum systems. Examples include topological codes, such as toric surface codes or Kitaev-like codes, which provide intrinsic protection against local errors by redundantly distributing quantum information over a specific topology. These codes are particularly promising for large qubit arrays where individual errors are common but manageable through topological redundancy [34, 35].

A concrete example is the surface code, which uses a two-dimensional grid of qubits to encode information. Each qubit is entangled with its neighbors, enabling the detection and correction of local errors. The specific topology of the surface code provides a natural barrier against local errors, allowing robust quantum computer operation despite inevitable physical errors.

The Surface Code is a type of topological quantum code that uses a two-dimensional grid of qubits to protect quantum information from local errors. Below is an example illustrating how this code works.

Suppose we have a 3 x 3 grid of qubits arranged as follows:

Q_{00}	Q_{01}	Q_{02}
Q_{10}	Q_{11}	Q_{12}
Q_{20}	Q_{21}	Q_{22}

Each qubit Q_{ij} interacts with its immediate neighbors through entangling operations, which are critical for error detection and correction.

The stabilizer of the surface code is defined by Pauli X and Z operators applied in specific configurations. For a 3 x 3 configuration, the stabilizer operators can be:

$$X_{00}X_{01}X_{10}X_{11}, \quad Z_{01}Z_{02}Z_{11}Z_{12}, \quad \text{etc.}$$

These operators enable the detection of local errors in the qubits. For example, if a qubit Q_{01} experiences an error and changes state, this will be reflected in the outcomes of the associated stabilizer measurements.

To detect errors, periodic measurements of the stabilizer operators are performed. Suppose that measuring $X_{00}X_{01}X_{10}X_{11}$ yields $+1$, indicating no error detected. However, measuring $Z_{01}Z_{02}Z_{11}Z_{12}$ yields -1, signaling the presence of an error in one of the neighboring qubits.

To correct the error, additional information from the stabilizer measurement outcomes is used to pinpoint the affected qubit and apply the appropriate correction

operation, which could be an X or Z operation depending on the context of the detected error.

The Surface Code provides robust protection against local errors by utilizing topological redundancy and entangling operations between neighboring qubits. While complex to implement, this approach offers a promising solution for building fault-tolerant quantum systems in the near future.

5 Discussion

The collection of information on noise correction in classical and quantum computing is fundamental to understanding the techniques and advancements in this field. For this purpose, the PRISMA methodology (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [36] was employed, offering a structured guide for conducting systematic reviews and meta-analyses.

The PRISMA methodology includes several key criteria:

- **Publication Date**: Priority was given to articles published in the last 10 years to ensure relevance and currency of the information.
- **Journal of Publication**: Articles from high-impact and prestigious journals in computing and physics, such as *Physical Review Letters*, *IEEE Transactions on Information Theory*, and *Nature*, were selected.
- **Keywords**: Keywords such as "quantum error correction," "classical error correction," "noise reduction," "quantum computing," and "error mitigation" were used to identify relevant studies.
- **Inclusion and Exclusion Criteria**: Studies addressing both the theory and practical implementation of error correction techniques in classical and quantum systems were included, while preliminary studies or those lacking significant results were excluded.

Classical error correction theory has been fundamental to the development of quantum error correction (QEC). In classical computing, error correction codes such as Hamming codes, Reed-Solomon codes, and convolutional codes have been widely used for detecting and correcting errors in data transmission and storage.

In the quantum context, QEC builds on similar principles but with greater complexity due to the quantum nature of information. Key classical methods and their quantum counterparts include:

- **Hamming Codes**: In the quantum realm, CSS (Calderbank-Shor-Steane) codes are equivalent to Hamming codes. They use two classical codes to correct Pauli X and Z errors.
- **Reed-Solomon Codes**: In quantum error correction, surface codes and color codes are equivalent to Reed-Solomon codes, offering enhanced robustness against more complex errors.

- **Convolutional Codes**: Quantum entanglement-based codes can be viewed as the quantum counterpart of convolutional codes, leveraging entanglement to correct distributed errors across multiple qubits.

These relationships highlight the importance of classical error correction theory as a foundation for developing error correction methods in quantum computing. Classical principles and techniques have been adapted and extended to address the unique challenges of quantum computing, such as decoherence and entanglement.

The implementation of error correction techniques is crucial for the effective development of software for quantum computers. Qubits, the basic elements of quantum computing, are extremely susceptible to errors due to interactions with their environment and the quantum operations themselves. Without error correction, the fidelity of quantum results rapidly degrades, affecting the practical utility of quantum computers.

Therefore, software developers for quantum computers must be aware of the need to incorporate error correction techniques into their algorithms and applications. This includes not only the use of error correction codes but also the implementation of error mitigation strategies. Integrating these techniques can significantly improve the accuracy and reliability of quantum results.

Additionally, quantum error correction enables the scalability of quantum systems. As quantum computers with larger numbers of qubits are developed, error correction will be essential to maintaining the coherence and functionality of these complex systems. Consequently, education and training in quantum error correction techniques are critical for quantum software developers.

Thus, the implementation of error correction techniques in the development of quantum computer software is vital to fully harness the potential of quantum computing and ensure precise and accurate results.

6 Conclusion

The review of the state of the art in the field of Quantum Error Correction (QEC) is essential to understanding and advancing one of the most critical areas of quantum computing. Classical computing theory, with its developments in error correction codes such as Hamming and Reed-Solomon codes, has provided the conceptual and mathematical foundation upon which quantum error correction is built. In quantum computing, computational errors are more prevalent due to the fragility of quantum states and decoherence, making error correction indispensable for the practical realization of functional and reliable quantum computers. This knowledge is not only vital for advancing the theory and practice of quantum computing but also crucial for training specialized scientists who can significantly contribute to this emerging field.

In Mexico, the training of specialized scientists in quantum error correction and quantum computing aligns perfectly with the strategic priorities of PRONACES

(National Strategic Programs) from Conahcyt. These programs emphasize the development of scientific and technological capacities that address the country's strategic needs. Promoting knowledge in quantum computing will enable Mexico to position itself at the forefront of this field, fostering an advanced scientific and technological culture. The training of experts in quantum error correction will not only strengthen the academic and research sectors but also create new opportunities in the technology and innovation industries, facilitating the development of advanced and competitive solutions in the global market.

In conclusion, quantum error correction is a key piece in the puzzle of quantum computing, enabling reliable and accurate computations in the presence of inherent errors. Mexico has the opportunity to lead in this field through the training of specialized scientists and the promotion of cutting-edge research aligned with PRONACES strategies. This effort will not only drive the country's technological development but also open new opportunities in areas such as quantum cryptography, material simulation, and the optimization of complex processes. The current status of quantum computing indicates that we are in a phase of rapid advancement and discovery, with a promising future where quantum error correction will play a fundamental role in unlocking the full potential of quantum technologies.

References

1. Mario Coccia: New directions of technologies pointing the way to a sustainable global society. Sustainable Futures, vol. 5, p. 100114, 2023. https://doi.org/10.1080/09537325.2022.2110056.
2. Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien: Quantum Error Correction. Presented at the Department of Materials, University of Oxford, Oxford, U.K., Quantum Motion, London, U.K., Google Quantum AI, Venice, CA, USA, NTT Computer and Data Science Laboratories, NTT Corporation, Musashino, Japan, Graduate School of China Academy of Engineering Physics, Beijing, China, 2023. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/2210.00921.
3. Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien: Quantum error mitigation. Rev. Mod. Phys., vol. 95, no. 4, pp. 045005, 2023. https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.95.045005.
4. S. Endo, S. C. Benjamin, and Y. Li: Practical Quantum Error Mitigation for Near-Future Applications. Phys. Rev. X, vol. 8, no. 3, pp. 031027, 2018. https://journals.aps.org/prx/abstract/10.1103/PhysRevX.8.031027.
5. R.P. Feynman: Simulating physics with computers. Int J Theor Phys, vol. 21, pp. 467–488, 1982. https://doi.org/10.1007/BF02650179.
6. R. W. Hamming: Error detecting and error correcting codes. The Bell System Technical Journal, vol. 29, no. 2, pp. 147–160, 1950. https://doi.org/10.1002/j.1538-7305.1950.tb00463.x.
7. F. Ivanov, E. Krouk, and V. Zyablov: New Code-Based Cryptosystem Based on Binary Image of Generalized Reed-Solomon Code. In: 2021 XVII International Symposium "Problems of Redundancy in Information and Control Systems".
8. P. S. Bere and M. Z. A. Khan: Low complexity, diversity preserving hard decision decoder for CRC codes with IoT applications. In: 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall), London, United Kingdom, 2022, pp. 1–5. https://doi.org/10.1109/VTC2022-Fall57202.2022.10012980.

9. M. Selvi, S. Jeeva, and J. Jaswanth: Improving the Power Efficiency in OFDM System using LDPC Codes. In: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 1717–1723. https://doi.org/10.1109/ICAAIC60222.2024.10575635.
10. A. L. Grimsmo and S. Puri: Quantum Error Correction with the Gottesman-Kitaev-Preskill Code. PRX Quantum, vol. 2, p. 020101, 2021. https://doi.org/10.1103/PRXQuantum.2.020101.
11. K. Noh, V. V. Albert, and L. Jiang: Quantum Capacity Bounds of Gaussian Thermal Loss Channels and Achievable Rates With Gottesman-Kitaev-Preskill Codes. IEEE Transactions on Information Theory, vol. 65, no. 4, pp. 2563–2582, 2019. https://doi.org/10.1109/TIT.2018.2873764.
12. D. Gottesman: Stabilizer codes and quantum error correction. Ph.D. dissertation, California Institute of Technology, 1997. ProQuest Dissertations and Theses, 9800357.
13. [[7, 1, 3]] Steane code. The Error Correction Zoo (V. V. Albert and P. Faist, eds.), 2024. https://errorcorrectionzoo.org/c/steane.
14. [[9, 1, 3]] Shor code. The Error Correction Zoo (V. V. Albert and P. Faist, eds.), 2022. https://errorcorrectionzoo.org/c/shornine.
15. V. Guruswami and M. Wootters: Repairing Reed-Solomon codes. In: 2016. ISBN: 9781450341325. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2897518.2897525.
16. Y. Gao, Q. Yue, X. Huang, and J. Zhang: Hulls of Generalized Reed-Solomon Codes via Goppa Codes and Their Applications to Quantum Codes. IEEE Transactions on Information Theory, vol. 67, no. 10, pp. 6619–6626, 2021. https://doi.org/10.1109/TIT.2021.3074526.
17. M. Moradi and A. Mozammel: Concatenated Reed-Solomon and Polarization-Adjusted Convolutional (PAC) Codes. In: 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Sofia, Bulgaria, 2022, pp. 123–128. https://doi.org/10.1109/BlackSeaCom54372.2022.9858284.
18. S. Vafi and P. H. Dung: Binary Burst Error Correcting Cyclic Codes Designed with the Circulant Parity Check Matrix. IEEE Communications Letters, vol. 16, no. 8, pp. 1277–1279, 2012. https://doi.org/10.1109/LCOMM.2012.052112.120768.
19. H. Praveena and K. Kalyani: FPGA implementation of Parity Check Matrix based Low Density Parity Check Decoder. In: 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 1214–1217. https://doi.org/10.1109/ICISC.2018.8398997.
20. M. Ellouze, R. Tajan, C. Leroux, C. Jégo, and C. Poulliat: Low-Complexity Algorithm for the Minimum Distance Properties of PAC Codes. In: 2023 12th International Symposium on Topics in Coding (ISTC), Brest, France, 2023, pp. 1–5. https://doi.org/10.1109/ISTC57237.2023.10273572.
21. O. Griebel, M. Herrmann, B. Hammoud, and N. Wehn: Energy-Efficient Decoding of Spatially Coupled Low-Density Parity-Check Codes using Adaptive Window Sizes. In: 2023 12th International Symposium on Topics in Coding (ISTC), Brest, France, 2023, pp. 1–5. https://doi.org/10.1109/ISTC57237.2023.10273561.
22. E. Aldabas, M. Corbalán, and J. M. Cisa: Introducción a los Códigos de Hamming. In: XII Jornades de Conferències JCEE'06, Escola Universitària de Enginyeria Tècnica Industrial de Terrassa, EUETIT "L'Escola Industrial", Colom, 1, 08222 TERRASSA, Barcelona, Universitat Politècnica de Catalunya (UPC), 2013.
23. S. Talmale, S. Unnikrishnan, and B. K. Lande, "Decoding Algorithm for Modified Linear Block Code with Syndrome Polynomial," in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, India, 2019, pp. 1–6. https://doi.org/10.1109/ICAC347590.2019.9036747.
24. A. Ashikhmin, C. -Y. Lai, and T. A. Brun, "Quantum Data-Syndrome Codes," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 3, pp. 449–462, 2020. https://doi.org/10.1109/JSAC.2020.2968997.

25. Rachel Won, "Optical signal processing: Efficient digitization," *Nature Photonics*, vol. 8, p. 423, 2014. doi: Not specified.
26. Erik J. Gustafson and Henry Lamm, "Robustness of Gauge Digitization to Quantum Noise," Fermi National Accelerator Laboratory, Batavia, Illinois, 60510, USA, Fermilab-Pub-23-018-SQMS-T, January 25, 2023.
27. J. Fan, M. Hsieh, H. Chen, H. Chen, and Y. Li, "Construction and Performance of Quantum Burst Error Correction Codes for Correlated Errors," in *2018 IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, 2018, pp. 2336–2340. https://doi.org/10.1109/ISIT.2018.8437493.
28. A. M. M. Zuhudi, K. Anwar, and G. Budiman, "On The Design of Non-CSS Quantum Error Correction Codes with High Quantum Information," in *2023 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)*, Denpasar, Indonesia, 2023, pp. 140–145. https://doi.org/10.1109/ICoABCD59879.2023.10390974.
29. Markus Grassl, "Quantum Error Correction," in *Quantum Information: From Foundations to Quantum Technology Applications*, Wiley, 2019, pp. 111–125. https://doi.org/10.1002/9783527805785.ch7.
30. T. C. Cuvelier, S. A. Lanham, B. R. L. Cour, and R. W. Heath, "Quantum Codes in Classical Communication: A Space-Time Block Code From Quantum Error Correction," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2383–2412, 2021. https://doi.org/10.1109/OJCOMS.2021.3121183.
31. A. Mondal and K. K. Parhi, "Quantum Circuits for Stabilizer Error Correcting Codes: A Tutorial," *IEEE Circuits and Systems Magazine*, vol. 24, no. 1, pp. 33–51, Firstquarter 2024. https://doi.org/10.1109/MCAS.2024.3349668.
32. G. S. Ravi et al., "VAQEM: A Variational Approach to Quantum Error Mitigation," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Seoul, Korea, Republic of, 2022, pp. 288–303. https://doi.org/10.1109/HPCA53966.2022.00029.
33. J. Zhang, Y.-X. Liu, R.-B. Wu, K. Jacobs, and F. Nori, "Quantum feedback: Theory, experiments, and applications," *Physics Reports*, vol. 679, pp. 1–60, 2017. https://doi.org/10.1016/j.physrep.2017.02.003. Available: https://www.sciencedirect.com/science/article/pii/S0370157317300479.
34. I. K. Sohn and J. Heo, "An Introduction to Fault-Tolerant Quantum Computation and its Overhead Reduction Schemes," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Prague, Czech Republic, 2018, pp. 44–46. https://doi.org/10.1109/ICUFN.2018.8436751.
35. N. Delfosse, B. W. Reichardt, and K. M. Svore, "Beyond Single-Shot Fault-Tolerant Quantum Error Correction," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 287–301, 2022. https://doi.org/10.1109/TIT.2021.3120685.
36. M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, D. Moher, "Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas," *Revista Española de Cardiología*, vol. 74, no. 9, pp. 790–799, 2021. https://doi.org/10.1016/j.recesp.2021.06.016. URL: https://www.revespcardiol.org/es-declaracion-prisma-2020-una-guia-articulo-S0300893221002748.
37. E. Berlekamp, R. McEliece and H. van Tilborg, "On the inherent intractability of certain coding problems (Corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978, https://doi.org/10.1109/TIT.1978.1055873.

Supervised Quantum Machine Learning

A. Monroy-Azpeitia, J. Y. Montiel-Pérez, A. Maldonado-Romo, A. A. Arvizu-Velazquez, J. L. López-Bonilla, and S. Vidal-Beltrán

Abstract The fusion between supervised learning and quantum computing promises substantial improvements in comparison with classic algorithms in the NISQ era, fuelling great interest for further developments in industry and academia. This chapter provides a review of the current state of quantum machine learning in supervised tasks, focusing on papers published in English in scientific journals or presented at conferences from January 2018 to June 2024, sourced from the SCOPUS database. Various papers were selected according to the PRISMA methodology that focus into general aspects of quantum supervised machine learning for variational algorithms and oracles, including theoretical case studies as well as examples and performance comparisons with classical versions on various datasets.

A. Monroy-Azpeitia (✉) · J. Y. Montiel-Pérez · A. A. Arvizu-Velazquez
Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico City, Mexico
e-mail: amonroya2025@cic.ipn.mx

J. Y. Montiel-Pérez
e-mail: yalja@ipn.mx

A. A. Arvizu-Velazquez
e-mail: aarvizuv2024@cic.ipn.mx

A. Maldonado-Romo
Quantum Open Source Foundation, Mexico City, Mexico
e-mail: alberto@qosf.net

J. L. López-Bonilla · S. Vidal-Beltrán
ESIME-Zacatenco, Instituto Politécnico Nacional, Mexico City, Mexico
e-mail: lopezb@ipn.mx

S. Vidal-Beltrán
e-mail: svidalb@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_10

1 Introduction

Quantum machine learning in supervised tasks represents a promising frontier that lies in the intersection of classical machine learning and quantum computing, enabling the handling of both classical and quantum data. Despite still being in an early stage of development and with QPUs lacking sufficient fault-tolerant qubits, significant advances in hybrid algorithms and techniques suggest a future where this emerging technology could address complex problems more efficiently than traditional classic methods. Further research and interdisciplinary collaboration are essential to address current challenges and unlock the full potential of this field, where quantum principles are integrated into classical machine learning algorithms to enhance performance. For instance, quantum Kernel-based algorithms are applied in supervised learning tasks and parametrized quantum circuits are utilized for classification and regression tasks. Approximation optimizers and quantum heuristics are examples of algorithms within the quantum optimization field. Quantum data processing, on the other hand, involves encoding classical data into quantum states and applying dimensionality reduction techniques.

Despite promising results in QML, technological advancements in building better QPUs with higher-quality resources and setting aside physical constraints quantum hardware development, there are well-defined problems that shape the field. Arbitrary classical data encoding, optimization of the training process and, notably, the lack of clarity on which types of real-world classical data benefit from QML over conventional classical algorithms.

Such challenges are being tackled by academia and the private sector through the use of real QPUs and simulators with on-going research aimed at identifying areas of interest where QML offers a genuine advantage over classical methods. A deeper understanding of QML could also lead to new insights in classical machine learning.

In this chapter we present the methodology used for a systematic review of research papers on QML in supervised tasks, the results obtained, a discussion from a digital transformation perspective and the conclusions.

2 Methodology

In this literature review, PRISMA methodology was used [1]. Two types of technical literature were considered, focusing only on works published in English. The first type include studies that contextualizes supervised learning algorithms alongside their quantum counter parts while explaining the foundations of quantum mechanics and quantum computing. Only research papers published in scientific journals that effectively condensate the topics mentioned were included. The second type comprises papers published in indexed journals between January 2018 and June 2024, distinguishing between those with a theoretical & design focus and those adopting

a more practical approach on specific datasets. Only papers from reputable journals, such as Quantum Intelligence, Nature, American Physical Society and IEEE were considered. Using SCOPUS as the search engine and database, a title-only search of published papers and conference proceedings was conducted, considering the use of relevant keywords and excluding works that did not provide significant advancements. Research was guided by the following questions:

- What are quantum machine learning algorithms in supervised learning tasks?
- What are the open-ended problems in this field?
- What ideas are being explored for performance improvement of this algorithms?
- On which datasets have these algorithms been applied?

3 Results

3.1 General Aspects of Supervised Quantum Machine Learning

3.1.1 Quantum Computing on the NISQ Era

The journey from Feynman's pioneering work in quantum computing [2, 3], which anticipated exponential speed improvements with quantum processors, to the development of several foundational quantum algorithms [4–7], has brought us to the NISQ (Noisy Intermediate-Scale Quantum) era [8]. In this phase, quantum computers are still not sufficiently fault-tolerant and lack the necessary number of qubits to fully achieve the breakthroughs promised by quantum supremacy, with only a few notable exceptions [9, 10].

In this context, the intersection of machine learning (ML) and quantum computing (QC) has given rise to Quantum Machine Learning (QML), a promising field aimed at accelerating supervised learning tasks for both classical and quantum datasets, offering potential advantages over traditional classical methods.

3.1.2 Supervised Quantum Machine Learning

One key difference between classical and quantum machine learning lies in the type of datasets they can process. Classical ML exclusively analyzes classical datasets, which contain no quantum state information. In contrast, QML can handle both classical and quantum datasets [11]. In the QML paradigm, supervised tasks define a dataset $\mathcal{D}$ as a Cartesian product of a feature set $\mathcal{X}$ and a label set $\mathcal{Y}$: $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$. For classification tasks, a binary quantum classifier comprises three main components [12]: an encoding process that maps numerical data $x \in \mathcal{X}$ into quantum states, a unitary operator that evolves the encoded data into new quantum states, and a decision

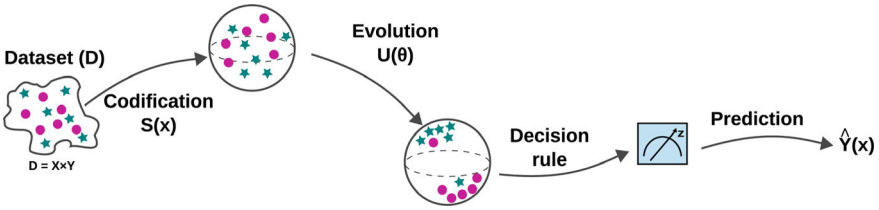


Fig. 1 Flowchart illustrating a quantum classifier for binary labeled data considering only a qubit: From left to right it starts with a dataset $D = \mathcal{X} \times \mathcal{Y}$, then a codification function of classical data maps it into a Hilbert space. After mapping, an evolution function $U(\theta)$ separates data and a decision rule that classifies the data through measurement that yields a final prediction $\hat{Y}(x)$

rule that classifies the resulting states. Figure 1 illustrates a one-qubit classification process for further explanation. Two prominent types of quantum algorithms for supervised QML are oracle-based algorithms and variational quantum algorithms. Variational algorithms use a hybrid training process to achieve class separability. In this process, the quantum circuit is optimized using classical ML techniques, while the user is responsible for designing and implementing the quantum circuit, decision rule, and dataset. The quantum computer executes the algorithm and adjusts the circuit parameters by minimizing a cost function [13]. Ensemble methods for quantum classifiers have also been explored and proposed [14].

In supervised QML with *Variational Quantum Circuits*, two prevalent hybrid methods are commonly used for the decision rule [15]: the *explicit method*, where the QPU learns a linear decision boundary in the associated Hilbert space based on qubit encoding, and the *implicit method*, which classifies data using kernel-based approaches, such as Support Vector Machines (SVM). In both methods, hyperparameter tuning is performed as a classical process.

3.1.3 Classical Data Encoding to Quantum States

Data encoding of classical data is a fundamental aspect of QML algorithms [16]. Examples of well-defined mapping architectures in the NISQ era include parametrized circuits that incorporate classical data for optimization purposes [17] and geometrically inspired mappings [18–20]. However, data encoding through parametrized quantum circuits can produce a maximally mixed state, which significantly reduces the efficiency of quantum classifiers [21]. QML models using this approach approximate continuous functions under typical encodings [22]. Additionally, research has explored data encodings tailored for kernel-based quantum classifiers [23].

3.1.4 Variational Quantum Circuits Optimization

Several factors influence the optimization (training) process of parametrized quantum circuits, including gradient vanishing, which increases with the number of qubits, entanglement, and noise in the circuits. These factors hinder convergence to the optimal solution [24, 25], revealing a relationship between the exponential concentration of cost values around the average and the exponential narrowing of minima [29]. This even suggests the impossibility of using QML for certain tasks, such as quantum scrambling [31]. To mitigate gradient vanishing, various parameter initialization strategies [28], including those leveraging transfer learning techniques, have been explored [27]. Additionally, barren plateaus in tensor network circuits exhibit characteristics of gradient vanishing [26, 30]. For kernel-based methods, the exponential concentration of kernels is understood as producing an initially data-independent model [31].

3.2 Quantum Support Vector Machines (QSVMs)

A kernel is defined as a mapping $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ for which a Hilbert space $\mathbb{H}$ and a characteristic mapping $\phi : \mathcal{X} \to \mathbb{H}$ exist such as:

$$K_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathbb{H}} \tag{1}$$

where $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ is the inner product defined over $\mathbb{H}$ and is a symmetrical and a positive semidefinite function.

In kernel methods [32–34], the dataset $\mathcal{X}$ is transformed into a high-dimensional Hilbert space $\mathbb{H}$ via a characteristic mapping ϕ. An inner product in this space is then used to create a linear classifier. For a binary classification task, a Support Vector Machine (SVM) seeks to construct a hyperplane that defines a decision boundary. For a data point $x \in \mathcal{X}$ and a binary label set $y \in \mathcal{Y} = \{\pm 1\}$, the classifier is defined as:

$$y(x) = \text{sgn}(\langle w, \phi(x) \rangle_{\mathbb{H}} + \beta) \tag{2}$$

where $\text{sgn}(a)$ is the sign function, which returns $+1$, -1, or 0 depending on whether a is positive, negative, or zero. Here, $w \in \mathbb{H}$ is the weight vector, and $\beta \in \mathbb{R}$ is the bias term. The representation theorem establishes a relationship between the kernel function and the SVM, expressing $w \in \mathbb{H}$ as a sum of mapped data points. The decision function is thus written as:

$$y(x) = \text{sgn}\left(\sum_{i \in S} \alpha_i \langle \phi(x_i), \phi(x) \rangle_{\mathbb{H}} + \beta \right) \tag{3}$$

where S is the index set of support vectors, which are the critical data points that influence the decision boundary.

The Q-SVM is built upon the mathematical formalism and axioms of quantum mechanics by considering the associated Hilbert space $\mathbb{H}$ as a *feature space* and generating the linear classifier from the inner product of $\mathbb{H}$. When the feature space grows and Kernel functions are hard to compute, the Q-SVM shows advantages over its classical counterpart [16]. The algorithm consists of applying a feature map $\hat{\mathcal{W}}(\mathbf{x})$ to the initial states $|0\rangle^{\otimes n}$ with a data dependency:

$$|\phi(\mathbf{x})\rangle = \hat{\mathcal{W}}(\mathbf{x})\,|0\rangle^{\otimes n} \tag{4}$$

where the kernel evaluation can be obtained by computing the fidelity between states, defined with the inner product as:

$$K_{ij} = \left\langle \phi(\mathbf{x_i}) | \phi(\mathbf{x_j}) \right\rangle \tag{5}$$

or by the projection of the quantum kernel:

$$K^{PQ}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \sum_{k=1}^{n} \|\rho_k(\mathbf{x}_i) - \rho_k(\mathbf{x}_j)\|_F^2\right) \tag{6}$$

with $\rho_k = \mathrm{tr}_{j \neq q} \rho_{total}$, i.e. the individual state of an only qubit in the density matrix formalism. Given a kernel function calculated using a quantum computer, an algorithm is implemented that leverages this kernel, while the training phase is carried out on a classical computer. In Fig. 2, a one-qubit example is presented. Several proposals for optimizing the Q-SVM include enhancing kernel quality through target-kernel alignment [35] and utilizing HHL-based Q-SVM algorithms for binary classification [36].

Explorations of potential applications of QSVMs have highlighted their use in various fields, particularly in the medical diagnosis of cancer [37–39], cardiac problems [40–43], and schizophrenia [44]. A more detailed review of the state-of-the-art applications of quantum machine learning algorithms in biomedicine can be found in [45]. Beyond biomedicine, QSVMs have been applied to solve the combinatorial

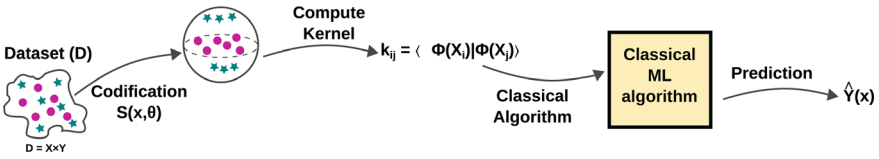


Fig. 2 Flowchart illustrating a Quantum Support Vector Machine (Q-SVM) classification task for one qubit that highlights the fidelity kernel computation process. For data set $D = \mathcal{X} \times \mathcal{Y}$, a codification function $S(x, \theta)$ that maps data. Then the kernel $k_{ij} = \phi(X_i)\phi(X_j)$, is computed and if followed by the use of a classical machine learning algorithm that yields a prediction

vehicle routing problem (VPR) [46], perform sentiment analysis [47], and detect fraud in financial transactions. In fraud detection, comparisons with classical methods like Random Forest and XGBoost demonstrate improved accuracy through the use of quantum feature selection [48].

3.3 Quantum Neural Networks

A Quantum Neural Network (QNN) is a neural network that combines quantum mechanics and classical neural networks and are used for processing either classical or quantum data with parametrized quantum circuits (PQC) which basically are a sequence of unitary transformations that depend on parameters that apply over input quantum states.

A basic QNN model for prediction tasks can be built by applying a quantum circuit consisting of encoded classical data $\hat{S}(x)$, a PQC $\hat{W}(\theta)$ with $\theta \in \mathbb{R}^k$:

$$\hat{U}(x, \theta) = \hat{W}(\theta)\hat{S}(x) \tag{7}$$

to the initial state of a quantum computer $|0\rangle^{\otimes n}$, therefore obtaining:

$$|\Psi(\theta, x)\rangle) = \hat{W}(\theta)\hat{S}(x)\,|0\rangle^{\otimes n} \tag{8}$$

where the expected values of the state are considered for prediction tasks of a classifier:

$$\mathbf{y}_\theta(x) = \left\langle \Psi(\theta, x)|\hat{\mathcal{M}}|\Psi(\theta, x)\right\rangle \tag{9}$$

this result is the expected value of the encoded and parametrized state of an observable $\hat{\mathcal{M}}$. Post-processing of the expected values correspond to a prediction of the model [49, 50]. In Fig. 3 a one qubit QNN is shown.

3.3.1 Q-LSTMs

Parametrized quantum circuits are used for building analog quantum models of recurrent neural networks such as the Long Short Term Memory (LSTM) [51]. In Fig. 4 an analogy of both models is shown. The quantum LSTM (Q-LSTM) uses PQC's instead of classical LSTM cells that result in a faster convergence, greater accuracy and stability [52] with quantum-inspired architectures [53]. Air pollution [54], sentiment analysis [55] and carbon price forecasting [56] are use cases in which Q-LSTM's are competing with their classical counterparts.

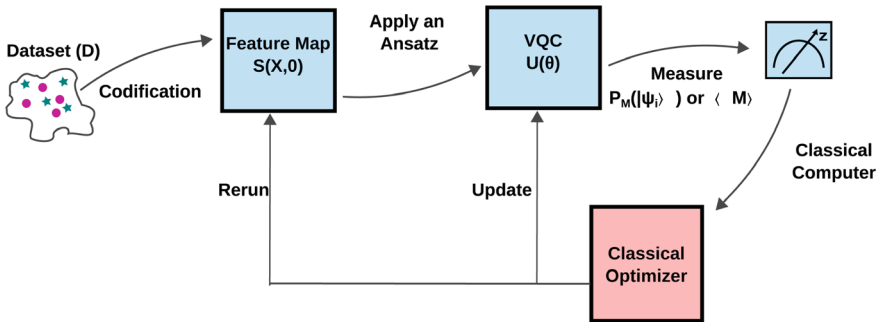


Fig. 3 Flowing illustrating the training process of a supervised quantum algorithm for binary classification using a feature map $S(X, 0)$ and a variational quantum circuit that uses an evolution function $U(\theta)$. The algorithm is the optimized by measuring the variational circuit and optimizing the previous steps classically

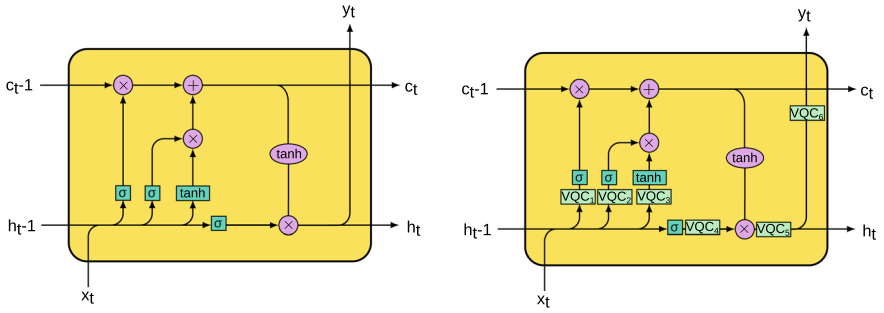


Fig. 4 Diagram comparing two neural network architectures. On the left side, a standard LSTM algorithm with inputs x_t, h_{t-1}, c_{t-1}, and outputs h_t, c_t. Inputs are activated via sigmoid σ and tanh activation functions. To the right, a Q-LSTM is shown where the main difference is that before processing inputs x_t, h_{t-1} with activation functions, they are mapped and processed through a variational quantum circuit that outputs a classic results that is then used as the input of the activation function

3.3.2 Tensor Networks

Originally used for many-body quantum system simulation focused on entanglement [57], tensor networks are used for big datasets approximations. A dataset, a quantum state or a higher-dimension multilinear map are decomposed and approximated with a multilinear map composition of smaller size [58]. Tensor networks are useful for decomposing higher dimensional data to a less complex structure which accelerates image classification algorithms [59, 60].

Tensor networks lie on the intersection of physics and machine learning, making them suitable for easy implementation on QPU's with models such as *tree tensor networks* (TTN) and *matrix product states* (MPS) [61]. In Fig. 5 TTN and MPS tensor architectures are shown of a tensor network.

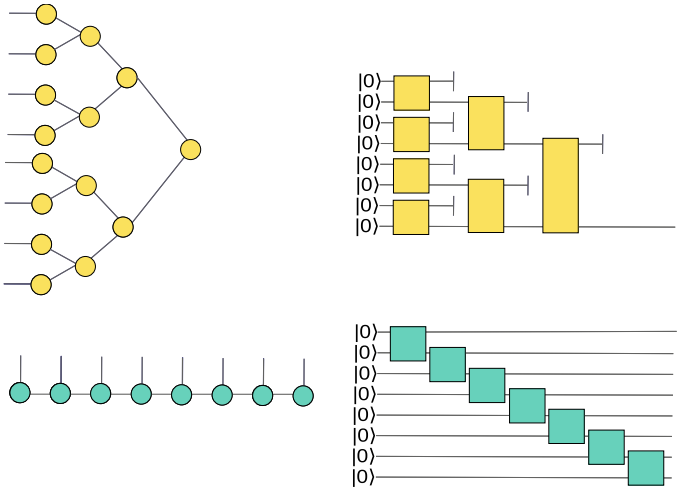


Fig. 5 Diagram illustrating the architecture of a tensor network and how is mapped to a quantum circuit. The top parts shows a basic TNN model of a tensor network and bottom part shows a basic MPS model of a tensor network

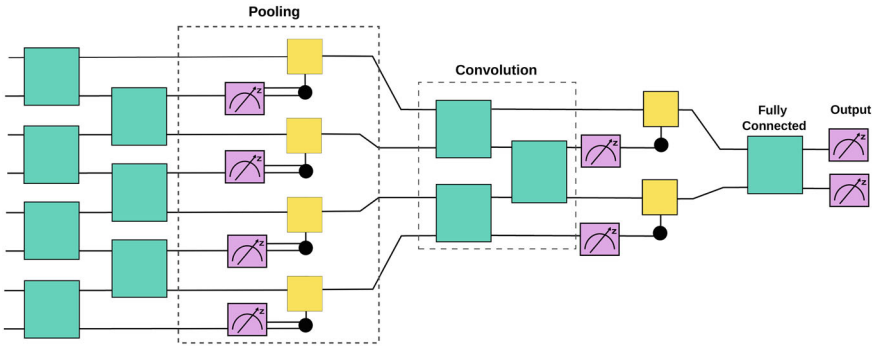


Fig. 6 Quantum circuit of a QCNN architecture. It begins with multiple input from TNN circuits, followed by pooling, convolution and fully connected layers that then are measured, obtaining a classic output

3.3.3 Quantum Convolutional Neural Networks (QCNN)

Classical convolutional neural networks (CNNs), which automatically identify relevant features without supervision [62] and are widely applied across various domains, serve as the inspiration for their quantum counterpart, Quantum Convolutional Neural Networks (QCNNs). As shown in Fig. 6, QCNNs feature an architecture analogous to classical CNNs, where the convolutional layer is implemented using quasi-local unitary operations. For pooling, intermediate measurements are performed on a fraction of the qubits in the circuit, with the results of these measurements used as inputs

for unitary rotations applied to nearby qubits. Convolutional and pooling layers are repeated until the system size is sufficiently reduced, after which a fully connected layer is implemented as a final unitary operation on all remaining qubits. The process concludes with a measurement on a fixed number of qubits. This quantum architecture is useful in recognizing quantum phases of matter for many-body quantum systems and in optimizing quantum error correction [63]. QCNNs are well-suited for the NISQ era due to their shallow circuit depth, which mitigates challenges such as barren plateaus [64].

Applications include analyzing high-energy physics datasets [65], for classifying and characterizing quantum phases of matter, and discovering order parameters that remain invariant under phase-preserving perturbations in quantum systems [66]. Recent studies have also shown their potential for image classification tasks [67] such as X-ray classification on the covid-cxr3 dataset, demostrating improved performance against CNN [68]. Similarly, they have been used for classification of RGB-images with a improved performace by using bigger ansatz [69] for the CIFAR-10 and MNIST datasets.

3.4 Quantum K-Nearest Neighbors

Inspired by the classical non-parametric supervised algorithm K-Nearest Neighbors (KNN), its quantum counterpart Q-KNN performs classification based on a defined distance metric. For classical data, this requires encoding into quantum states or a quantum random access memory (QRAM) encoding [70]. It shares fundamental similarities with classical RAM, enabling access to multiple memory cells simultaneously through superposition. This feature is particularly useful when it serves as a subroutine within a larger quantum algorithm. Grover's search algorithm solves non-structured search tasks for a size N set and reduces time complexity to $O(\sqrt{N})$ and recent Q-KNN inspired algorithms pair it with Hamming's metric for accuracy improvement of a recommendation system. In [71], the Mahalanobis distance is stored between states in the QRAM, and sampling from the collection is used to obtain the K Nearest Neighbors via Grover's algorithm. In [72], a new similarity measure called *polar distance* is proposed, combined with a classical binary search and Grover's algorithm. In [73], the Euclidean distance is used with Amplitude Estimation (AE) instead of QRAM, achieving a query complexity proportional to $O(\sqrt{M \log(M)})$. QRAM and Grover's algorithm, alongside angle encoding are proposed as a new framework for data encoding. Potential applications include incorporating the Hamming distance and designing specialized oracles to enhance accuracy and efficiency [74].

4 Discussion

Quantum Machine Learning (QML) in supervised tasks combines classical machine learning techniques with the formalism of quantum mechanics, offering significant advancements, particularly in areas like classical data encoding into quantum states. QML uses quantum properties such as superposition and entanglement to reduce resource consumption, including the number of required qubits. Data processing strategies often depend on classical optimization algorithms or complex schemes like dynamic systems, depending on the chosen algorithmic model—whether parametrized quantum circuits or oracle-based algorithms. Despite the complexity of big-data processing, hybrid classical-quantum methods, such as QRAM, are employed to improve K-NN computations. Rotation operators open the door to more sophisticated data encoding methods, including those based on multi-qubit entanglement, which are central to Q-SVM architectures. Quantum Neural Networks (QNNs) aim to simplify transitions between layers while looking to match or exceed classical performance.

Despite recent progress, significant problems remain, including optimization of training processes and arbitrary classical data encoding, which require further research efforts to overcome. identifying real-world use cases where quantum algorithms demonstrate clear superiority over classical methods remains an ongoing area of exploration.

5 Conclusion

Still in its developmental stage, supervised quantum machine learning has the potential to revolutionize complex data problems. As more powerful quantum computers are developed, deeper integration of quantum algorithms into practical applications is expected. These advancements are particularly evident in the innovation of data encoding techniques and the adoption of hybrid algorithms, which promise to enhance efficiency and performance. However, significant challenges, such as qubit limitations and noise, persist and demand ongoing research and development.

As a result of our investigation, a strong interest in exploring diverse datasets to apply supervised QML algorithms, aiming for improvements over classical counterparts is evident. A need for a deeper understanding of QML leads to new strategies and methods for practical applications in the NISQ era that pave the way for new research areas.

Interdisciplinary research and the development of innovative techniques to address current challenges are a necessity. Close collaboration between academia and industry is essential for testing and refining new algorithms. It not only advances quantum machine learning but also holds the potential to provide significant discoveries in classical algorithms, deepening the mathematical connections between the two approaches.

In conclusion, while the path to a practical implementation of quantum machine learning is challenging, the potential benefits fully justify sustained efforts in this field. With the continuous advancement of quantum technology and the development of more effective methodologies, we can anticipate a future where quantum algorithms play a pivotal role in solving complex problems across various scientific and technological disciplines.

References

1. Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hrobjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *Research Methods & Reporting* 372 :n71. (2021). https://doi.org/10.1136/bmj.n71
2. Feynman, R.P.: Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 467–488 (1982). https://doi.org/10.1007/BF02650179
3. Feynman, R.P.: Quantum mechanical computers. *Foundations of Physics*, 16, 507–531 (1986). https://doi.org/10.1007/BF01886518
4. Bennett, C.H., Brassard, G.: Quantum Cryptography: Public Key Distribution and Coin Tossing. *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, 175–179 (1984). https://doi.org/10.1016/j.tcs.2014.05.025
5. Deutsch, D.: Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London A*, 400(1818), 97–117 (1985). https://doi.org/10.1098/rspa.1985.0070
6. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
7. Grover, L.K.: A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, Philadelphia, Pennsylvania, USA, 212–219 (1996). https://doi.org/10.1145/237814.237866
8. Preskill, J.: Quantum Computing in the NISQ era and beyond. *Quantum*, 2, 79 (2018). https://doi.org/10.22331/q-2018-08-06-79
9. Arute, F., Arya, K., Babbush, R., et al.: Quantum supremacy using a programmable superconducting processor. *Nature*, 574, 505–510 (2019). https://doi.org/10.5061/dryad.k6t1rj8
10. Madsen, L.S., Laudenbach, F., Askarani, M.F., et al.: Quantum computational advantage with a programmable photonic processor. *Nature*, 606, 75–81 (2022). https://doi.org/10.1038/s41586-022-04725-x
11. Khan, T.M., Robles-Kelly, A.: Machine Learning: Quantum vs Classical. *IEEE Access*, 8, 219275–219294 (2020). https://doi.org/10.1109/ACCESS.2020.3041719
12. LaRose, R., Coyle, B.: Robust data encodings for quantum classifiers. *Physical Review A*, 102, 032420 (2020). https://doi.org/10.1103/PhysRevA.102.032420
13. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4, 043001 (2020). https://doi.org/10.1088/2058-9565/ab4eb5
14. Abbas, A., Schuld, M., Petruccione, F.: On quantum ensembles of quantum classifiers. *Quantum Machine Intelligence*, 2 (2020). https://doi.org/10.1007/s42484-020-00018-6
15. Schuld, M., Killoran, N.: Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, 122, 040504 (2019). https://doi.org/10.1103/PhysRevLett.122.040504

16. Havlíček, V., Córcoles, A.D., Temme, K., et al.: Supervised learning with quantum-enhanced feature spaces. *Nature*, 567, 209–212 (2019). https://doi.org/10.1103/PhysRevA.106.042431
17. Hubregtsen, T., Wierichs, D., Gil-Fuster, E., Derks, P.H.S., Faehrmann, P.K., Meyer, J.J.: Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, 106, 042431 (2022).
18. Larocca, M., Sauvage, F., Sbahi, F.M., Verdon, G., Coles, P.J., Cerezo, M.: Group-invariant quantum machine learning. *PRX Quantum*, 3, 030341 (2022). https://doi.org/10.1103/PRXQuantum.3.030341
19. Meyer, J.J., Mularski, M., Gil-Fuster, E., Mele, A.A., Arzani, F., Wilms, A., Eisert, J.: Exploiting symmetry in variational quantum machine learning. *PRX Quantum*, 4, 010328 (2023). https://doi.org/10.1103/PRXQuantum.4.010328
20. Glick, J.R., Gujarati, T.P., Corcoles, A.D., Kim, Y., Kandala, A., Gambetta, J.M., Temme, K.: Covariant quantum kernels for data with group structure. *Nature Physics*, 20, 479–483 (2024). https://doi.org/10.1038/s41567-023-02340-9
21. Li, G., Ye, R., Zhao, X., Wang, X.: Concentration of Data Encoding in Parameterized Quantum Circuits. *Advances in Neural Information Processing Systems*, Volume 35 (2022).
22. Goto, T., Tran, Q.H., Nakajima, K.: Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces. *Physical Review Letters*, 127, 090506 (2021). https://doi.org/10.1103/PhysRevLett.127.090506
23. Suzuki, Y., Yano, H., Gao, Q., Uno, S., Tanaka, T., Akiyama, M., Yamamoto, N.: Analysis and synthesis of feature map for kernel-based quantum classifier. *Quantum Machine Intelligence*, 2 (2020). https://doi.org/10.1007/s42484-020-00020-y
24. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9, 4812 (2018). https://doi.org/10.1038/s41467-018-07090-4
25. Arrasmith, A., Cerezo, M., Czarnik, P., Cincio, L., Coles, P.J.: Effect of barren plateaus on gradient-free optimization. *Quantum*, 5, 558 (2021). https://doi.org/10.22331/q-2021-10-05-558
26. Holmes, Z., Arrasmith, A., Yan, B., Coles, P.J., Albrecht, A., Sornborger, A.T.: Barren Plateaus Preclude Learning Scramblers. *Physical Review Letters*, 126, 190501 (2021). https://doi.org/10.1103/PhysRevLett.126.190501
27. Liu, H.Y., et al.: Mitigating barren plateaus with transfer-learning-inspired parameter initializations. *New Journal of Physics*, 25, 013039 (2023). https://doi.org/10.1088/1367-2630/acb58e
28. Kashif, M., Rahid, M., Al-Kuwari, S., Shafique, M.: Alleviating Barren Plateaus in Parameterized Quantum Machine Learning Circuits: Investigating Advanced Parameter Initialization Strategies. *Design, Automation and Test in Europe Conference and Exhibition*, Valencia (2024). https://doi.org/10.23919/DATE58400.2024.10546644
29. Arrasmith, A., et al.: Equivalence of quantum barren plateaus to cost concentration and narrow gorges. *Quantum Science and Technology*, 7, 045015 (2022). https://doi.org/10.1088/2058-9565/ac7d06
30. Cervero Martín, E., Plekhanov, K., Lubasch, M.: Barren plateaus in quantum tensor network optimization. *Quantum*, 7, 974 (2023). https://doi.org/10.22331/q-2023-04-13-974
31. Thanasilp, S., Wang, S., Cerezo, M., et al.: Exponential concentration in quantum kernel methods. *Nature Communications*, 15, 5200 (2024). https://doi.org/10.1038/s41467-024-49287-w
32. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning*, 20, 273–297 (1995). https://doi.org/10.1007/BF00994018
33. Vapnik, V.N.: The Support Vector method. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.D.: *Artificial Neural Networks - ICANN'97. Lecture Notes in Computer Science*, vol 1327. Springer, Berlin, Heidelberg (1997). https://doi.org/10.1007/BFb0020124
34. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support Vector Regression Machines. *Advances in Neural Information Processing Systems*, 9 (2003).

35. Hubregtsen, T., Wierichs, D., Gil-Fuster, E., Derks, P.J.H.S., Faehrmann, P.K., Meyer, J.J.: Training Quantum Embedding Kernels on Near-Term Quantum Computers. *Physical Review A*, 106, 042431 (2022). https://doi.org/10.1103/PhysRevA.106.042431
36. Nohara, T., Oyama, S., Noda, I.: Integrating Machine Learning Algorithms With Quantum Annealing Solvers for Online Fraud Detection. *IEEE Access*, 10 (2022). https://doi.org/10.1109/ACCESS.2022.3190897
37. Vashisth, S., Dhall, I., Aggarwal, G.: Design and analysis of quantum powered support vector machines for malignant breast cancer diagnosis. *Journal of Intelligent Systems*, 30(1), 998–1013 (2021). https://doi.org/10.1515/jisys-2020-0089
38. Yu, J.Y., Sim, W.S., Jung, J.Y., Park, S.H., Kim, H.S., Park, Y.R.: Evaluation of conventional and quantum computing for predicting mortality based on small early-onset colorectal cancer data. *Applied Soft Computing*, 162 (2024). https://doi.org/10.1016/j.asoc.2024.111781
39. Aswiga, R.V., Sridevi, S., Indira, B.: Leveraging Quantum Kernel Support Vector Machine for breast cancer diagnosis from Digital Breast Tomosynthesis images. *Quantum Machine Intelligence*, 6 (2024). https://doi.org/10.1007/s42484-024-00170-3
40. Prabhu, S., Gupta, S., Prabhu, G.M., Dhanuka, A.V., Bhat, K.V.: QuCardio: Application of Quantum Machine Learning for Detection of Cardiovascular Diseases. *IEEE Access*, 11, 136122–136135 (2023). https://doi.org/10.1109/ACCESS.2023.3338145
41. Babu, S.V., Ramya, P., Gracewell, J.: Revolutionizing heart disease prediction with quantum-enhanced machine learning. *Scientific Reports*, 14 (2024). https://doi.org/10.1038/s41598-024-55991-w
42. Enad, H.G., Mohammed, M.A.: Cloud computing-based framework for heart disease classification using quantum machine learning approach. *Journal of Intelligent Systems*, 33 (2024). https://doi.org/10.1515/jisys-2023-0261
43. Ozpolat, Z., Karabatak, M.: Performance Evaluation of Quantum-Based Machine Learning Algorithms for Cardiac Arrhythmia Classification. *Diagnostics*, 13 (2023). https://doi.org/10.3390/diagnostics13061099
44. Aksoy, G., Cattan, G., Chakraborty, S., Karabatak, M.: Quantum Machine-Based Decision Support System for the Detection of Schizophrenia from EEG Records. *Journal of Medical Systems*, 48 (2024). https://doi.org/10.1007/s10916-024-02048-0
45. Maheshwari, D., Garcia-Zapirain, B., Sierra-Sosa, D.: Quantum Machine Learning Applications in the Biomedical Domain: A Systematic Review. *IEEE Access*, 10 (2022). https://doi.org/10.1109/ACCESS.2022.3195044
46. Mohanty, N., Behera, B.K., Ferrie, C.: Solving the vehicle routing problem via quantum support vector machines. *Quantum Machine Intelligence*, 6 (2024). https://doi.org/10.1007/s42484-024-00161-4
47. Ruskanda, F.Z., Abiwardani, M.R., Mulyawan, R., Syafalni, I., Larasati, H.T.: Quantum-Enhanced Support Vector Machine for Sentiment Classification. *IEEE Access*, 11 (2023). https://doi.org/10.1109/ACCESS.2023.3304990
48. Grossi, M., Ibrahim, N., Radescu, V., Loredo, R., Voigt, K., Von Altrock, C., Rudnik, A.: Mixed quantum-classical method for fraud detection with quantum feature selection. *IEEE Transactions on Quantum Engineering*, 3 (2022). https://doi.org/10.1109/TQE.2022.3213474
49. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. *Phys. Rev. A*, 101, 032308 (2020). https://doi.org/10.1103/PhysRevA.101.032308
50. Schuld, M., Petruccione, F.: *Machine Learning with Quantum Computers*. Springer, 2nd Ed. (2021).
51. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
52. Chen, S. Y.-C., Yoo, S., Fang, Y.-L. L.: Quantum Long Short-Term Memory. In Proceedings of the 47th IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2022-May, pp. 8622–8626. (2022). https://doi.org/10.1109/ICASSP43922.2022.9747369
53. Rivera-Ruiz, M.A., Mendez-Vazquez, A., López-Romero, J.M.: Time Series Forecasting with Quantum Machine Learning Architectures. In: Pichardo Lagunas, O., Martínez-Miranda, J., Martínez Seis, B. (eds) Advances in Computational Intelligence. MICAI 2022. Lecture Notes

in Computer Science, vol 13612. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19493-1
54. Zheng, K., Van Griensven, J., Fraser, R.: A Quantum Machine Learning Approach to Spatiotemporal Emission Modelling. *Atmosphere*, 14, 944 (2023). https://doi.org/10.3390/atmos14060944
55. Chu, Z., Wang, X., Jin, M., Zhang, N., Gao, Q., Shao, L.: An Effective Strategy for Sentiment Analysis Based on Complex-Valued Embedding and Quantum Long Short-Term Memory Neural Network. *Axioms*, 13, 207 (2024). https://doi.org/10.3390/axioms13030207
56. Cao, Y., Zhou, X., Fei, X., Zhao, H., Liu, W., Zhao, J.: Linear-layer-enhanced quantum long short-term memory for carbon price forecasting. *Quantum Machine Intelligence*, 5(2) (2023). https://doi.org/10.1007/s42484-023-00115-2
57. Orús, R.: A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349, 117–158 (2014).
58. Sengupta, R., Adhikary, S., Oseledets, I., Biamonte, J.: Tensor networks in machine learning. *Eur. Math. Soc. Mag.*, 126, 4–12 (2022). https://doi.org/10.4171/MAG/101
59. Selvan, R., Dam, E.B.: Tensor Networks for Medical Image Classification. In: Proceedings of the Third Conference on Medical Imaging with Deep Learning, PMLR 121:721–732 (2020).
60. Zhu, C., Wang, L., Zhao, W.: Image classification based on tensor network DenseNet model. *Appl Intell*, 54, 6624–6636 (2024). https://doi.org/10.1007/s10489-024-05472-4
61. Huggins, W., Patil, P., Mitchell, B., Whaley, K.B., Stoudenmire, E.M.: Towards Quantum Machine Learning with Tensor Networks. *Quantum Science and Technology*, 4, 024001 (2019). https://doi.org/10.1088/2058-9565/aaea94
62. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al.: Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377 (2018). https://doi.org/10.1016/j.patcog.2017.10.013
63. Cong, I., Choi, S., Lukin, M.D.: Quantum convolutional neural networks. *Nat. Phys.*, 15, 1273–1278 (2019). https://doi.org/10.1038/s41567-019-0648-8
64. Pesah, A., Cerezo, M., Wang, S., Volkoff, T., Sornborger, A.T., Coles, P.J.: Absence of Barren Plateaus in Quantum Convolutional Neural Networks. *Phys. Rev. X*, 11 (2021). https://doi.org/10.1103/PhysRevX.11.041011
65. Chen, S.Y., Wei, T.C., Zhang, C., Yu, H., Yoo, S.: Quantum Convolutional Neural Networks for High Energy Physics Data Analysis. *Phys. Rev. Res.*, 4 (2022). https://doi.org/10.1103/PhysRevResearch.4.013231
66. Liu, Y.J., Smith, A., Knap, M., Pollmann, F.: Model-Independent Learning of Quantum Phases of Matter with Quantum Convolutional Neural Networks. *Phys. Rev. Lett.*, 130 (2023). https://doi.org/10.1103/PhysRevLett.130.220603
67. Shi, S., Wang, Z., Li, J., Li, Y., Shang, R., Zhong, G.: Quantum convolutional neural networks for multiclass image classification. *Quantum Inf. Process.* 23 (2024). https://doi.org/10.1007/s11128-024-04360-7
68. Yousif, M., Al-Khateeb, B.: A New Quantum Circuits of Quantum Convolutional Neural Network for X-Ray Images Classification. *IEEE Access*, 12 (2024). https://doi.org/10.1109/ACCESS.2024.3396411
69. Wu, C., Fu, W., Hu, W., Li, X., Yang, Y., Li, Y., Jing, Y.: RGB image classification with quantum convolutional ansatz. *Quantum Inf. Process.* 21 (2022). https://doi.org/10.1007/s11128-022-03442-8
70. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum Random Access Memory. *Phys. Rev. Lett.* 100, 160501 (2008). https://doi.org/10.1103/PhysRevLett.100.160501
71. Gao, L.Z., Lu, C.Y., Guo, G.D., Zhang, X., Lin, S.: Quantum K-nearest neighbors classification algorithm based on Mahalanobis distance. *Front. Phys.* 10, 1047466 (2022). https://doi.org/10.3389/fphy.2022.1047466
72. Feng, C., Zhao, B., Zhou, X., Ding, X., Shan, Z.: An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance. *Entropy* 25, 127 (2023). https://doi.org/10.3390/e25010127

73. Wiebe, N., Kapoor, A., Svore, K.M.: Quantum Nearest-Neighbor Algorithms for Machine Learning. *Rinton Press. Quantum Inf. Comput.* (2015).
74. Maldonado-Romo, A., Montiel-Pérez, J.Y., Onofre, V., Maldonado-Romo, J., Sossa-Azuela, J.H.: Quantum K-Nearest Neighbors: Utilizing QRAM and SWAP-Test Techniques for Enhanced Performance. *Mathematics* (2024). https://doi.org/10.3390/math12121872
75. Li, J., Lin, S., Yu, K., et al.: Quantum K-nearest neighbor classification algorithm based on Hamming distance. *Quantum Inf Process* 21, 18 (2022). https://doi.org/10.1007/s11128-021-03361-0

Hybrid Quantum Genetic Algorithm with a Strongly Entanglement Recombination Process for Maximizing the Coverage Area of a Set of Unmanned Aerial Vehicles in Natural Disaster Locations

Enrique Ballinas, Oscar Montiel, and Anabel Martínez-Vargas

Abstract This paper introduces a Hybrid Quantum Genetic Algorithm (HQGA-strong), which utilizes strongly entangled quantum states in the recombination process to enhance the coverage area of unmanned aerial vehicles (UAVs) in disaster-affected areas. The HQGA-strong is a fusion of quantum computing benefits and traditional methods, creating a quantum population that improves the recombination process. Moreover, a fuzzy inference system is implemented to determine the most suitable angle for the quantum rotation gate, instead of a conventional lookup table or mathematical equation. To showcase the advantages of quantum genetic algorithms in terms of accuracy and convergence time compared to classical genetic algorithms, various statistical tests were conducted. Depending on the data distribution, both parametric and non-parametric tests were utilized. The results demonstrated that the HQGA-strong performed comparably in accuracy to its quantum counterpart. The hybrid nature of the algorithm is highlighted by its operation: initialization and measurement processes are conducted on a quantum computer, while the remainder of the process runs on a classical computer. Two experimental scenarios were designed, termed "best case" and "worst case," each assessing different aspects of algorithmic performance.

These authors contributed equally to this work.

E. Ballinas (✉)
Faculty of Chemical Sciences and Engineering, Universidad Autónoma de Baja California, Tijuana, Baja California, Mexico
e-mail: luis.ballinas@uabc.edu.mx

O. Montiel
Department, Centro de Investigación y Desarrollo de Tecnología Digital, Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

A. Martínez-Vargas
Research, Innovation and Graduated Department, Universidad Politécnica de Pachuca, Pachuca, Hidalgo, Mexico
e-mail: anabel.martinez@upp.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_11

Keywords Hybrid quantum algorithms · Quantum-inspired algorithms · Unmanned aerial vehicles · Fuzzy logic

1 Introduction

A natural disaster can be described as a calamitous event that, unfortunately, often results in the loss of thousands of human lives. In addition, the infrastructure of the place where the natural disaster strikes can be heavily damaged, generating significant expenses on certain occasions. Events such as hurricanes [1–3], earthquakes [4], or landslides [5] are generated by the environment rather than human-driven. Therefore, when the natural disaster is over, it is imperative to re-establish a line of communication to help the injured population and thus prevent the number of victims from increasing.

Presently, some approaches such as an integrated communication system based on the internet protocol for a post-disaster emergency rescue communication platform [1], a methodology to design an effective and comprehensive set of warning messages to decrease the number of victims in a flood situation [6], or a framework that uses graph-theoretic topological robustness metrics to evaluate the network resilience against natural disasters [7] have been implemented. Although the aforementioned proposals prove their effectiveness against natural disasters, the use of UAVs as aerial base stations (BSs) to provide communication in the affected area after a catastrophe has received significant attention due to their low cost, high mobility, cost-effectiveness, and flexible deployment [8, 9]. This problem is considered an NP-hard problem since it requires many possible configurations to position the UAVs to maximize the coverage area in the disaster zone [10].

Various classical methodologies have been proposed to address the challenge of positioning UAVs as aerial base stations in natural disaster scenarios. One notable approach is the use of classical meta-heuristics. These algorithms are particularly suited to this issue as they are capable of generating high-quality solutions to complex problems within polynomial time [11]. We define classical meta-heuristics as algorithms that operate on traditional (conventional) computers. In the existing literature, the majority of studies approach the problem of UAV placement in disaster zones as a path planning challenge [12–18]. Nevertheless, there are alternative strategies that tackle this problem differently. For instance, some utilize a (1+1) evolution strategy (ES) [19], while others combine multiple classical meta-heuristics [20, 21], employ computer vision techniques [22], or develop heuristics for joint planning [23].

The works mentioned above utilize classical meta-heuristics to solve NP-hard problems, such as the problem of maximizing the coverage area provided by a set of UAVs in a disaster zone, as addressed in this paper. However, we hypothesize that the aforementioned results can be enhanced through the application of quantum computing.

In quantum computing, several intriguing properties such as superposition, quantum entanglement, and quantum parallelism stand out. Due to these characteristics,

quantum computing serves as a highly effective methodology for solving optimization problems. An evolutionary algorithm, a type of meta-heuristic, utilizes operators based on the principles of biological evolution to develop a set of solutions (a population). Furthermore, the unique properties of quantum computing can be leveraged to enhance the performance of evolutionary algorithms, enabling them to effectively solve NP-hard problems such as the Knapsack problem (KP) [24], the Traveling Salesman Problem (TSP) [25], and the challenge of maximizing the coverage area provided by a set of UAVs [26].

Path planning is a key focus of many studies on quantum evolutionary algorithms to locate UAVs in natural disaster areas [27–31]. Some other studies explore the problem of the pickup and delivery of a set of UAVs [32, 33], finding an optimal set of routes for intercluster UAV communication [34], processing high-dimensional data in homeland security imaging systems [35], and updating the positions of a set of terrestrial robots in a search and rescue scenario [36]. None of the aforementioned studies address the problem of maximizing the coverage area provided by a set of UAVs in a disaster zone using a hybrid quantum genetic algorithm; this paper is detailed in [26]. Actually, the hybrid quantum algorithm presented in this paper is an updated version of the quantum algorithm shown in [26]. The main differences include the use of a strongly entanglement quantum circuit for the recombination process, and the parameters in the fuzzy inference system used to update the rotation angle of the quantum rotation gates.

The main contribution of this paper to state-of-the-art technology is the integration of a quantum recombination operator based on a strongly entanglement circuit. The inclusion of quantum features provides the algorithm with several quantum advantages, including quantum parallelism, quantum entanglement, as well as quantum haploid recombination and mutation, which all contribute to an enhancement of the algorithm's performance. With classical computing methods, it is impossible to achieve these attributes. Moreover, the experimental results when compared with those of a classical genetic algorithm and a hybrid quantum genetic algorithm demonstrated superior average solutions to the task to maximize the coverage area in a disaster zone given a set of UAVs, providing significant results to the literature.

The paper is organized as follows: Sect. 2 introduces the main theoretical concepts involved in the study. Section 3 details the structure of the proposed HQGA-strong to maximize the coverage area in a disaster zone given a set of UAVs. Section 4 discusses the results of the experiments conducted when the HQGA-strong was implemented in the IBM Qiskit simulator. Finally, Sect. 5, summarizes our findings and suggests directions for future research.

2 Theoretical Framework

Quantum computing is a field that harnesses principles of quantum mechanics, such as entanglement, superposition, and quantum parallelism to design quantum algorithms [37]. These characteristics increase the potential of quantum algorithms com-

pared to classical algorithms (algorithms implemented in non-quantum computers), since quantum computers process information millions of times faster than a classical computer [38]. Therefore, the design of quantum algorithms is becoming an interesting area to solve non-polynomial time problems.

2.1 Quantum Computing

A qubit in a non-basic state can represent two possible values simultaneously. This characteristic is called superposition. The superposition principle is defined as the linear combination of two or more state vectors that creates a new state vector in the same Hilbert space [37].

To describe the state of a qubit the Dirac notation is used. Mathematically, a qubit, say $|\psi\rangle$, is represented as a matrix of complex numbers c_0 and c_1,

$$|\psi\rangle = c_0\,|0\rangle + c_1\,|1\rangle \tag{1}$$

where $|c_0|^2 + |c_1|^2 = 1$ according to the Born rule [39]. Thus, $|c_0|^2$ and $|c_1|^2$ represent the probability of finding the qubit after being measured in the states $|0\rangle$ and $|1\rangle$, respectively [40]. A set of two or more qubits is known as a quantum register. A quantum register of n-qubits on the computational basis ($|0\rangle$,$|1\rangle$) can be found in superposition in any of the possible 2^n states $|00...0\rangle$, $|00...1\rangle$, $|11...1\rangle$. In general, the state of a quantum register of two qubits is represented as,

$$|\psi\rangle = c_0\,|00\rangle + c_1\,|01\rangle + c_2\,|10\rangle + c_3\,|11\rangle \tag{2}$$

where $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$. This property allow us to handle many bit strings in a single quantum register.

The general form to represent a quantum register of n-qubits is

$$|\psi\rangle = c_0\,|00\ldots0\rangle + c_1\,|00\ldots1\rangle + \ldots + c_{2^n-1}\,|11\ldots1\rangle = \sum_{i=0}^{2^n-1} c_i\,|i\rangle \tag{3}$$

where $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$ and $|i\rangle$ represent the eigenstates of the computational base whose bit values are equivalent to the decimal number expressed in base notation 2.

2.2 Strongly Entanglement

In quantum computation, quantum entanglement occurs when a quantum state cannot be decomposed into the tensor product of its individual states [40]. This indicates that the states are inseparable. An example of this is a 2-qubit entangled state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{4}$$

If we express Eq. 4 in the form of a tensor product, the resulting expression is as follows:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}. \tag{5}$$

In this context, a_0, a_1 and b_0, b_1 denote the amplitude values for the first and second qubits, respectively.

According to Eq. 5 the conditions $a_0 \neq 0$, $a_1 \neq 0$, $b_0 \neq 0$ and $b_1 \neq 0$ must be met to satisfy $a_0 b_0 = 1/\sqrt{2}$ and $a_1 b_1 = 1/\sqrt{2}$. However, this leads to a contradiction as $a_0 b_1$ results in 0, despite the previously mentioned requirement that a_0 and b_1 be non-zero. Consequently, the quantum state demonstrates entanglement, as it cannot be decomposed into individual tensor products of its constituent states.

To measure the degree of entanglement in a 2-qubit quantum state, we employ a metric known as tangle. Tangle is defined as the square of the concurrence [41], which for a 2-qubit state, $|\psi\rangle$, is expressed by the equation:

$$Concurrence(|\psi\rangle) = |\langle\psi| \left|\tilde{\psi}\right\rangle|. \tag{6}$$

Here, $\tilde{\psi}$ represents the spin-flipped version of $|\psi\rangle$ and is defined as $\tilde{\psi} = (Y \otimes Y)\,|\psi*\rangle$. Here, Y is the Pauli-Y matrix and $\tilde{\psi}$ is $|\psi\rangle$ with its amplitude complex conjugated. For an unentangled state, the concurrence is zero, whereas for a maximally entangled state, such as a Bell state, the concurrence reaches one.

2.3 Disaster Zone Model

Figure 1 illustrates the natural disaster scenario discussed in this paper. The simulation models the area as a two-dimensional space confined within a square of $L \times L$ m^2 [42]. The simulated area has its origin in (0,0) and it is limited by $[X_{max}, -X_{max}]$ and $[Y_{max}, -Y_{max}]$. The set of UAVs is defined as $M = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where (x_i, y_i) is the position of the UAV in two-dimensional space. Here, i is the index of the set of UAVs ranging from 1 to m, where $m = |M|$. The set of mobile users MUs is $N = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where (x_i, y_i) is the cartesian coordinate of the MU. Similarly to the set of UAVs, i is the index of the set of MUs ranging from 1 to n, where $n = |N|$ [42].

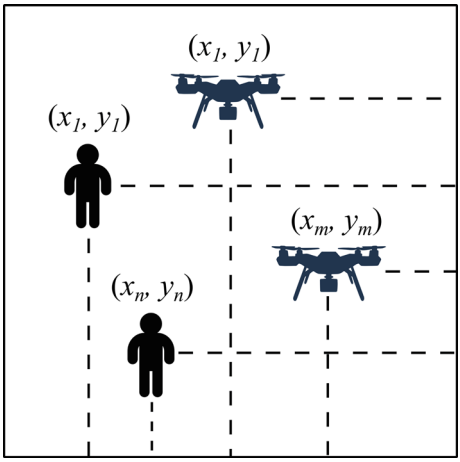


Fig. 1 Illustration of n mobile users MUs and m UAVs within a simulated natural disaster scenario

2.4 Objective Functions and Restrictions

The objective function, as defined in Eq. 7 calculates the ratio of mobile users (MUs) covered by a set of UAVs, denoted by C [42, 43]

$$\text{Maximize } C = \frac{v}{n}. \tag{7}$$

From Eq. 7, v represents the number of mobile users (MUs) covered by a set of UAVs, while n denotes the total number of MUs in the disaster area, with $(n \neq 0)$.

To ascertain whether a mobile user (MU) is within the coverage area of a UAV, a set of constraints must be met. The primary constraint is the path loss, calculated using the Hata propagation model for urban areas at an operational frequency of 700 MHz. The path loss Lp between the i-th UAV and the j-th MU is given by

$$Lp_{((x_i,y_i),(x_j,y_j))}(dB) = A + B\log_{10}(d_{((x_i,y_i),(x_j,y_j))}) \tag{8}$$

where A and B are constants. $d_{((x_i,y_i)(x_j,y_j))}$ represents the distance between the Cartesian coordinates of the i-th UAV and the j-th MU.

The next constraint is the signal interference ratio (SIR), which indicates the level of interference an MU perceives when a set of MUs transmits over the same channel. The SIR for the j-th MU is given by [42]:

$$SIR_{(x_j,y_j)}(dB) = Ptx_{(x_i,y_i)} + Lp_{((x_i,y_i),(x_j,y_j))} - I \tag{9}$$

where $Ptx_{(x_i,y_i)}$ is the transmission power of the i-th UAV. $Lp_{((x_i,y_i),(x_j,y_j))}$ is the path loss from the i-th UAV to the j-th MU in the desired signal. I is the total interference and is given by:

$$I = \sum_{k \in K} Ptx_{(x_k, y_k)} - d^{\delta}_{((x_k, y_k),(x_j, y_j))} \tag{10}$$

where $Ptx_{(x_k, y_k)}$ is the transmission power of the k-th MU interferer. To indicate the MU transmitting over the same channel as the j-th MU within the desired signal, we use the index k. This strategy is known as spectrum sharing [44]. The range of k is $0 \leq k \leq |K|$, where K represents the set of interferers, meaning the set of MUs transmitting on the same channel. The term $d^{\delta}_{((x_k, y_k),(x_j, y_j))}$ denotes the Euclidean distance between the k-th interferer and the j-th MU. Here, δ is the attenuation factor, taking a value between 2 and 4.

In general, to associate an i-th UAV with a j-th MU, the constraints outlined in the equations from 11 to 14 must be satisfied [42]. According to Eq. 11, the j-th MU must fall within the coverage area of the i-th UAV. Similarly, Eq. 12 specifies the permissible path loss between the i-th UAV and the j-th MU, ensuring that it stays within a predefined threshold. Equations 11 and 12 establish thresholds for mitigating interference based on exhaustive simulations. Additionally, Eq. 13 requires that the SIR at the j-th MU be maintained above 3 dB for successful voice transmission. Finally, Eq. 14 ensures the exclusive association of the j-th MU with the i-th UAV, where $0 < l \leq m$, $0 < i \leq m$, and $0 \leq j \leq n$.

$$d_{((x_i, y_i),(x_j, y_j))} \leq 2000\, m, \tag{11}$$

$$Lp_{((x_i, y_i),(x_j, y_j))} \leq 120 dB, \tag{12}$$

$$SIR_{(x_j, y_j)} \geq 3 dB, \tag{13}$$

$$N_j \in M_i \therefore N_j \notin M_l \; \forall l \neq i. \tag{14}$$

3 Methodology

An HQGA-strong algorithm for positioning UAVs in strategic zones to maximize network coverage in a disaster area is presented in this section. The HQGA-strong employs a strong entanglement process as a recombination and mutation method, similar to the techniques used in classical genetic algorithms. Furthermore, this quantum circuit updates the quantum population through quantum rotation gates. These rotation gates, with adaptive rotation angles, refine the final solutions to enhance the convergence of the HQGA-strong.

The quantum circuit of the HQGA-strong is illustrated in Fig. 2. This circuit utilizes one of the most powerful features of quantum computation: entanglement. Through quantum entanglement, the HQGA-strong employs a quantum register of ten qubits, enabling it to generate a population of 1024 (2^{10}) quantum chromosomes. In contrast, a classical genetic algorithm would require 1024 classical registers to achieve the same. This results in an exponential reduction in memory storage, thereby

increasing the search space and allowing the generation of more possible solutions with just a few qubits.

A strongly entanglement quantum circuit, as shown in Fig. 2 is characterized by a set of controlled-NOT gates (C-NOT gates). In Fig. 2, the C-NOT gates are enclosed within a square dotted line. A C-NOT gate is a 2-qubit quantum gate that operates with two inputs: the control qubit and the target qubit. The control qubit (black dot) determines the output state of the target qubit. For instance, in Fig. 2, if the inputs are $q_1 = q_2 = |0\rangle$, then the output of q_2 remains the same as its input. Conversely, if $q_1 = |1\rangle$ the output of q_2 will be the opposite of its input. Therefore, a C-NOT gate in Fig. 2 can be expressed as $|q_1, q_2\rangle = |q_1, q_1 \oplus q_2\rangle$, where $\oplus$ denotes the binary exclusive OR operation [40].

In Fig. 2, the C-NOT gates are connected sequentially. Specifically, the qubit q_1 serves as the control qubit and q_2 as the target qubit for the first C-NOT gate. For the second C-NOT gate, q_2 becomes the control qubit, and q_3 is the target qubit. This sequence continues in this manner, except for the final C-NOT gate. The last C-NOT gate is connected in reverse: the qubit q_m acts as the control qubit, while q_1 is the target qubit. This configuration propagates quantum entanglement to the subsequent layers of the quantum circuit. For this quantum circuit, only one strong entanglement layer was used. The C-NOT gates are responsible for generating strong entanglement within the quantum circuit, as they are among the quantum gates that exhibit maximal entanglement power. The entanglement power of a 2-qubit gate, U, denoted as EP(U), is defined as the mean tangle generated by U, averaged over all input product states uniformly sampled on the Bloch sphere [41]. Mathematically, this is expressed as:

$$EP(U) = \langle E(U\,|\psi_1\rangle \otimes |\psi_2\rangle)\rangle_{|\psi_1\rangle,|\psi_2\rangle} \tag{15}$$

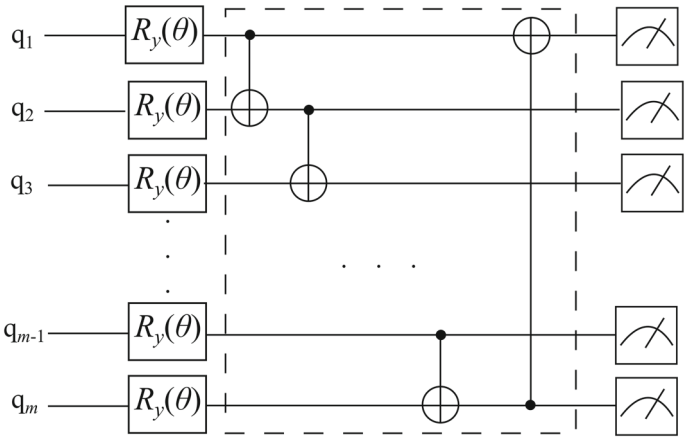


Fig. 2 Quantum circuit of the HQGA-strong implemented in the IBM Qiskit simulator. The square dotted line indicates the strong entanglement layer

where E represents the tangle, which can be any 2-qubit entanglement measure, such as linear entropy. The states $|\psi_1\rangle$ and $|\psi_2\rangle$ are single-qubit states uniformly sampled on the Bloch sphere.

At the beginning of the quantum circuit of the HQGA-strong (see Fig. 2) there are quantum rotation gates ($R_y(\theta)$). These gates set the quantum register into a superposition state, enabling the generation of new quantum individuals before applying the strong entanglement layer. Finally, measurement operators are used to produce a classical population.

Quantum circuit length is defined as the number of quantum gates (quantum operations) within the circuit when represented as a sequence of operations. Fundamentally, it measures the depth of the circuit, indicating the number of computations required to execute a quantum algorithm. Shorter quantum circuits are generally more efficient and less complex. Therefore, circuit length serves as a key indicator of a quantum circuit's complexity [41].

Algorithm 1 describes the process of the HQGA-strong to compute the maximum coverage area provided by a set of UAVs in a disaster zone. The inputs to the HQGA-strong include the problem parameters, the maximum number of generations, MAX_GEN, and the number p of quantum chromosomes. In the experiments, a single quantum chromosome was used. The output of the HQGA-strong is the maximum coverage area, denoted by C. Similar to classical evolutionary algorithms, the HQGA-strong begins by initializing a generation counter t. In step 2 of Algorithm 1, the HQGA-strong creates the scenario by positioning N mobile users in the affected area.

Algorithm 1 Hybrid Quantum Genetic Algorithm with a strongly entanglement recombination process.

Require: The number p of quantum chromosomes, the maximal number of generations MAX_GEN, the total number of mobile users N, and the total number of UAVs M.
Ensure: Maximum coverage C as shown in Eq. 7
1: $t \leftarrow 0$
2: Randomly generate the positions of N
3: Initialize Q(t) using the strongly entanglement circuit and R_y quantum rotation gates
4: **while** $t <$ MAX_GEN **do**
5: Measure Q(t) using Eq. 17 to generate P(t)
6: Map P(t) with random UAVs' coordinates. A UAV_P(t) population is created.
7: Evaluate UAV_P(t) in the fitness function using Eq. 7
8: Create a new quantum population Q(t) using the strongly entanglement recombination process
9: Update the adaptive rotation angle θ of the quantum rotation gate Ry using a fuzzy inference system.
10: Store the best solutions b of UAV_P(t) in B(t)
11: $t = t + 1$
12: **end while**

The initial quantum population Q(t) is created in step 3 of Algorithm 1. A quantum population Q(t) is a set of quantum registers represented as Q(t) =

$\left[\left|\Psi_1^t\right\rangle, \left|\Psi_2^t\right\rangle, \ldots, \left|\Psi_p^t\right\rangle\right]$ where t is the generation counter and p is the size of the population. A quantum chromosome is defined as:

$$\left|\Psi_i^t\right\rangle = \left[\left|\psi_{i,1}^t\right\rangle \left|\psi_{i,2}^t\right\rangle \ldots \left|\psi_{i,m}^t\right\rangle\right] \tag{16}$$

where m is the number of qubits, and $i = 1, 2, \ldots, p$. Here, $\left|\psi_{i,1}^t\right\rangle = \alpha_{i,1} + \beta_{i,1}$; $\left|\psi_{i,2}^t\right\rangle = \alpha_{i,2} + \beta_{i,2}$; $\ldots$; $\left|\psi_{i,m}^t\right\rangle = \alpha_{i,m} + \beta_{i,m}$, where $|\alpha|^2 + |\beta|^2 = 1$. Therefore, $|\alpha|^2$ and $|\beta|^2$ represent the probabilities of finding the qubit in the state $|0\rangle$ and $|1\rangle$, respectively, after measurement. The length of a quantum chromosome is equivalent to the number of UAVs. For each quantum chromosome $\left|\Psi_i^t\right\rangle$, the following operations are performed (see Fig. 2): First, the Qiskit simulator initializes the quantum register in a zero basis state as follows: $\left|\Psi_i^t\right\rangle = [\left|0_{i,1}^t\right\rangle \left|0_{i,2}^t\right\rangle \ldots \left|0_{i,m}^t\right\rangle]$. Next, a set of $R_y(\theta)$ quantum rotation gates are applied. Since the initial value of θ is zero, the initial state of the quantum register does not change; thus, the strong entanglement layer cannot create quantum entanglement between the qubits.

Step 4 of Algorithm 1 encompasses a loop that includes steps 5 to 12. This loop terminates when the maximum number of generations, MAX_GEN, is reached. In step 5, a measurement operator is applied, resulting in the creation of a classical population $P(t)$ according to $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. In quantum mechanics, an observable is represented by a Hermitian operator that can be expressed through spectral decomposition: $O = \sum \lambda_i P_i$, where λ_i denotes the eigenvalues of the operator O, i represents any given state, and P_i are the projectors associated with the eigenspaces of O. The operator O is decomposed into its eigenvalues and corresponding projectors. A projector represents the probability of transitioning to an eigenvector and is characterized by idempotence and orthogonality, collectively summing to the identity, $\sum P_i = I$ [45]. By applying Born's rule, the probability measurement λ_i is computed as follows: $\mathbf{P}(\lambda_i) = \langle\psi|\, P_i\, |\psi\rangle$ [46]. The post-measurement state, $\left|\psi'\right\rangle$, is then calculated as follows:

$$\left|\psi'\right\rangle = \frac{P_i\,|\psi\rangle}{\sqrt{\langle\psi|\, P_i\, |\psi\rangle}}. \tag{17}$$

In step 6 of Algorithm 1, random UAV coordinates $M = (x_1, y_1), \ldots, (x_m, y_m)$ are generated. The classical chromosomes of the population $P(t)$ are then associated with a UAV coordinate (x_m, y_m) (see Fig. 3). At this point, a population of UAV coordinates, UAV_P(t), is generated. In Fig. 3, the bits with a value of 1 indicate that the m-th UAV is selected; otherwise, the m-th UAV is not present in the affected area. The UAV_P(t) population represents the positions of UAVs in the disaster area.

The UAV_P(t) population is evaluated in the fitness function (step 7) according to Algorithm 2, proposed by [43]. In Algorithm 2, the UAV counter m is initialized in step 1. This counter indicates the index of a specific UAV. The loop in step 2 iterates until all UAVs have been verified. Step 3 involves initializing the counter for mobile users, n. Step 4 contains a conditional that checks whether a UAV is selected ($U_m = 1$) or not ($U_m = 0$). If the conditional is true, i.e., $U_m = 1$, the loop in step 5 traverses through each of the mobile users, MU, and step 6 continues to check if the n-th MU (associated mobile user) is not already associated with any UAV U_m. Next,

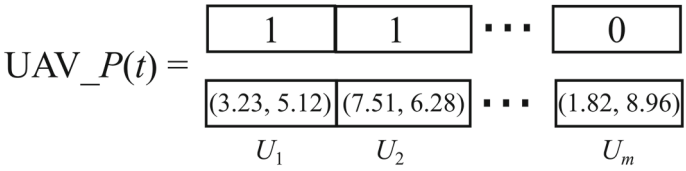


Fig. 3 Mapping process between random UAV coordinates and the population $P(t)$. In the figure, the population $P(t)$ is represented by zeros and ones (upper part), while the UAV positions are indicated by the numbers in the lower part

step 7 calculates the Euclidean distance d_{nm} between U_m and MU_n. If d_{nm} satisfies the constraints in Eq. 11 (step 8), the path loss Lp_{nm} is calculated (step 9). As long as Lp_{nm} meets the conditions outlined in Eq. 12 (step 10), the SIR of MU_n is computed in step 11. The MU_n is associated with U_m in step 13 if its SIR complies with Eq. 13 (step 12). Once the associated user MU_n covered by U_m is added, the fitness value C (see Eq. 7) can be determined. We decided to implement the fitness function outlined in Eq. 7, as the main objective of our HQGA-strong is to demonstrate that it can generate a higher coverage rate than a classical genetic algorithm. For that reason, Eq. 7 considers the ratio of mobile users covered by a set of UAVs to the total number of mobile users in the disaster area.

Algorithm 2 Fitness function to evaluate UAV_$P(t)$

Require: Number of mobile users N, number of UAVs M
Ensure: Maximum coverage C as shown in Eq. (7)

```
1: m = 0
2: while m < M do
3:     n = 0
4:     if U_m = 1 then
5:         while n < N do
6:             if MU_n = 0 then
7:                 Compute Euclidean distance d_nm between U_m and MU_n
8:                 if d_nm ≤ 2 then
9:                     Compute pathloss Lp_nm
10:                    if Lp_nm < 120dB then
11:                        Compute SIR of the MU_n
12:                        if SIR_n ≥ 3dB then
13:                            Associate MU_n to U_m
14:                        end if
15:                    end if
16:                end if
17:            end if
18:            n = n + 1
19:        end while
20:    end if
21:    Add associated MU_n
22:    m = m + 1
23: end while
24: Calculate C
```

From Algorithm 1, step 8 is executed, where a new quantum population is created using a strong entanglement quantum circuit. This quantum circuit functions as a recombination process, similar to classical genetic algorithms. In classical genetic algorithms, single-point and double-point crossover recombination processes are the most commonly used [47]. The key difference between these classical methods and the quantum recombination method proposed in this work is that, in classical methods, one or two crossover points are randomly selected from the parent chromosomes. In contrast, the quantum recombination method generates superposition and quantum entanglement across the entire population using the R_y and CNOT gates, thereby creating a new population in each generation. The evolutionary aspect of HQGA-strong is maintained by preserving the best solution from the previous generation in the current population. This ensures that HQGA-strong continues its search based on previous solutions rather than starting from scratch. The quantum circuit used as a recombination process in this paper was proposed by [48].

In step 9 of Algorithm 1, the adaptive rotation angle θ is updated by a fuzzy inference system (FIS). Here, the term adaptive refers to the rotation angle's ability to adjust according to the output generated by the HQGA-strong. The FIS consists of one input and one output. The input variable in the FIS is the coverage percentage (C) computed by the fitness function (see Eq. 7), where zero represents 0% and one represents 100%. Here, there are eight linguistic terms: NC, VLC, LC, SC, AC, GC, VGC, and OC. The output variable in the FIS is the rotational angle θ. It also includes eight linguistic terms: DAL, DL, DEL, DELM, DNDI, IL, IAL, and IAL2. The universe of discourse for the output is measured in radians, ranging from -4π (–10) to 4π (10). This range is chosen because, in classical computing, a rotation of 2π is required to return an object to its original state. In contrast, in quantum computing, a rotation of 4π is necessary to restore a qubit to its original state. This phenomenon is known as Dirac's belt trick or Feynman's plate trick [41].

Tables 1 and 2 present the types and parameter values of membership functions (MFs) and the fuzzy rules of the FIS, respectively. In Table 1, the bold letters indicate the inputs (C), while the others represent the outputs (θ). For example, if C is VLC (very low coverage), then θ is IAL (increase a lot). For the inputs, NC, VLC, LC, SC, AC, GC, VGC, and OC correspond to no coverage, very low coverage, low coverage, stable coverage, acceptable coverage, good coverage, very good coverage, and optimal coverage, respectively. For the outputs, IAL2, IAL, IL, DAL, DL, DEL, DELM, and DNDI represent increase a lot 2, increase a lot, increase a little, decrease a lot, decrease long, decrease a little, decrease a little more, and do not decrease or increase. The difference between IAL2 and IAL lies in the type of membership function and the range. Once the adaptive rotation angle θ is calculated, it is applied to the quantum rotation gate R_y, which is mathematically represented as follows:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{18}$$

Finally, steps 10 and 11 of Algorithm 1 are executed sequentially. In step 10, the best solution is stored in the set of best solutions, denoted as $B(t)$. Then, the

Table 1 Characteristics of the FIS. In the first column, the variable name indicates the input (C) and output (θ) of the fuzzy system. The second column lists the term names assigned to each membership function. The third column represents the shape of the membership functions, and the fourth column shows the range of the universe of discourse for each membership function

Variable name	Term name	Type	Parameters
C	NC	Trapezoidal	[0.0, 0.1, 0.2]
C	VLC	Triangular	[0.1, 0.2, 0.3]
C	LC	Triangular	[0.2, 0.3, 0.4]
C	SC	Triangular	[0.3, 0.4, 0.5]
C	AC	Triangular	[0.4, 0.5, 0.6]
C	GC	Triangular	[0.5, 0.6, 0.7]
C	VGC	Triangular	[0.6, 0.7, 0.8]
C	OC	Trapezoidal	[0.7, 0.8, 1,1]
θ	DAL	Trapezoidal	$[-4\pi, -4\pi, -3\pi, -2\pi]$
θ	DL	Triangular	$[-3\pi, -2\pi, -\pi]$
θ	DEL	Triangular	$[-2\pi, -\pi, -0.5\pi]$
θ	DELM	Triangular	$[-\pi, -0.5\pi, 0]$
θ	DNDI	Triangular	$[-0.5\pi, 0.3, \pi]$
θ	IL	Triangular	$[0, \pi, 2\pi]$
θ	IAL	Triangular	$[\pi, 2\pi, 3\pi]$
θ	IAL2	Trapezoidal	$[2\pi, 3\pi, 4\pi, 4\pi]$

Table 2 Fuzzy rule matrix of the FIS

NC	VLC	LC	SC	AC	GC	VGC	OC
IAL2	IAL	IL	DAL	DL	DEL	DELM	DNDI

generation counter t is incremented (step 11) until the maximum generation criterion, denoted as MAX_GEN, is met. The quantum computer executes steps 3, 5, and 8; meanwhile, the rest of the steps are carried out by a classical computer.

4 Experiments and Results

To test the performance of the Hybrid Quantum Genetic Algorithm with a strongly entanglement recombination process (HQGA-strong), comparisons were made against a Hybrid Quantum Genetic Algorithm with Fuzzy Adaptive Rotation Angle (HQGAFARA) and a classical genetic algorithm (GA). The HQGAFARA follows the same quantum and classical principles as our HQGA-strong, as shown in Algorithm 1. The primary difference between the two quantum algorithms lies in the recombination process. HQGAFARA utilizes a Deutsch-Jozsa quantum circuit to

recombine the quantum population, and then employs quantum R_y rotation gates to mutate the recombined quantum population. A more detailed explanation of HQGA-FARA is provided in [26]. Another important distinction is the complexity between the two quantum algorithms, specifically regarding the execution time of the quantum circuit on the quantum computer. A longer quantum circuit is considered more complex; thus, the length of a quantum circuit is a primary indicator of complexity, defined as the minimum number of computational steps required for quantum gates to be applied in parallel to implement the circuit [41]. The quantum circuit created for HQGAFARA is longer than that implemented in HQGA-strong, indicating that the complexity of the quantum circuit in our proposed HQGA-strong is more efficient. To maximize the coverage area of a set of unmanned aerial vehicles in a natural disaster location, the following parameters were configured in the GA: double-point crossover recombination method, tournament selection, crossover rate of 0.5, and a population size of 100 chromosomes for the initial experiments.

A set of three experiments was conducted to evaluate the performance of the proposed hybrid quantum genetic algorithm. The first set assesses the overall performance of the quantum genetic algorithms compared to the classical genetic algorithm; here, we present the best, worst, and average solutions. Additionally, the standard deviation is provided to demonstrate the robustness of each algorithm.

The second set of experiments involves statistical tests regarding accuracy. These tests evaluate whether there is a significant difference between the quantum and classical algorithms in terms of accuracy, utilizing parametric or non-parametric methods based on the data generated by each algorithm.

The third set of experiments presents statistical tests concerning convergence time. In these experiments, we focus on the number of generations each algorithm requires to find the best solution in each run, rather than execution time. For the second and third set of experiments, we decided to use statistical tests, as these tools provide us with a rigorous framework to analyze and improve the performance of the algorithms described in this paper.

The three sets of experiments were divided into two cases:

- Best case scenario. In this case, multiple transmission channels were utilized to assist the affected area.
- Worst case scenario. In this scenario, only one transmission channel was used to assist the affected area, presenting a challenge for the proposed quantum genetic algorithm.

In both cases, experiments were conducted with five (M = 5), ten (M = 10), and twenty (M = 20) UAVs, along with a total of 500 mobile users (N = 500) in the disaster area. In all experiments, 30 runs were performed, each consisting of 1000 generations.

The emergency network was designed using the values presented in Table 3. The parameters were obtained from [43]. The transmission power of a mobile user is represented by PT_n, while the transmission power of each UAV is represented by PT_m. A UAV's altitude U_m and a mobile user's MU_n are the values that result in

Table 3 Parameters used in the emergency LTE mobile network

Parameter	Values
MU_n transmit power PT_n	24 dBm
U_m transmit power PT_m	73 dBm
Altitude of U_m	150 m
Height of MU_n	1.5 m
Frequency f_c	700 MHz

lower path losses. Furthermore, f_c represents the frequency that will be used by Public Safety Communications (PSC) to provide LTE broadband communications.

Table 4 presents the results of the overall performance comparison between the quantum genetic algorithms and the classical genetic algorithm in the best case and worst case scenario. The first column (M) indicates the number of UAVs deployed in the simulated disaster zone, which spans 100 km^2. The second column details the algorithms employed to maximize the coverage area of unmanned aerial vehicles in a natural disaster setting. The key distinction, GA-32 and GA-100, lies in their population sizes, with 32 and 100 chromosomes, respectively. Similarly, GA-1024 utilizes a population of 1024 chromosomes. Given that the quantum genetic algorithms (HQGAFARA and HQGA-strong) generate a quantum population of 32 chromosomes with five UAVs, using five qubits to create $2^5 = 32$ possible configurations, we matched this population size in the GA for a fair comparison. The results from Table 4 section Best case scenario reveal higher accuracy with HQGAFARA and HQGA-strong compared to the GA with 32 chromosomes, as indicated by the mean solutions. Moreover, the quantum genetic algorithms demonstrate greater robustness, evidenced by their lower standard deviations. With a population of 100 chromosomes, the GA potentially benefits from a larger search space, enhancing solution diversity. However, the experimental outcomes affirm the superior precision and accuracy of HQGAFARA and HQGA-strong over GA-100. This superiority remains consistent with 10 (M = 10) and 20 (M = 20) UAVs. The fifth and sixth columns of Table 4 display the worst and standard deviation of the solutions, respectively, with the quantum algorithms outperforming the GA in both scenarios-whether the GA population is 32 or 100 chromosomes. From Table 4 section Worst case scenario, the accuracy and robustness of HQGAFARA and HQGA-strong are better than GA since both quantum algorithms obtained higher mean solutions and their Std is lower than GA.

In the previous paragraph, we compared quantum genetic algorithms against a classical genetic algorithm. Now, we will analyze the performance differences between HQGAFARA and HQGA-strong as presented in Table 4. With five UAVs, HQGA-strong outperforms HQGAFARA as evidenced by its higher mean solution and lower standard deviation, demonstrating greater accuracy and precision. However, the situation changes with 10 and 20 UAVs. In both scenarios, HQGAFARA slightly outperforms HQGA-strong, with a marginal difference of 0.02 (2%). For instance, with 10 UAVs, HQGAFARA covers approximately 300 mobile users out

Table 4 Experimental results for the best case and worst case scenario. M represents the number of UAVs in the disaster zone. Std means Standard deviation. The best results are in bold

		Best case scenario				Worst case scenario			
M	Algorithm	Best	Mean	Worst	Std	Best	Mean	Worst	Std
5	GA-32	0.36	0.32	0.29	0.015	**0.18**	0.15	0.13	0.0103
	GA-100	0.36	0.34	0.32	0.009	0.17	0.15	0.15	0.0053
	HQGAFARA	0.36	0.35	0.34	0.006	0.17	0.165	0.15	**0.0050**
	HQGA-strong	**0.36**	**0.36**	**0.34**	**0.004**	0.174	**0.168**	**0.154**	0.0072
10	GA-1024	0.59	0.53	0.5	0.019	0.26	0.24	0.23	0.0087
	GA-100	0.58	0.54	0.47	0.026	0.27	0.25	0.22	0.0133
	HQGAFARA	**0.6**	**0.6**	**0.57**	**0.009**	0.27	0.265	0.25	**0.0039**
	HQGA-strong	0.59	0.58	0.55	0.014	**0.276**	**0.271**	**0.26**	0.0061
20	GA-1024	0.79	0.71	0.68	0.02	0.36	0.33	0.31	0.0122
	GA-100	0.8	0.75	0.7	0.02	**0.39**	0.35	0.32	0.0164
	HQGAFARA	**0.81**	**0.8**	**0.79**	0.01	0.38	**0.368**	**0.36**	0.0052
	HQGA-strong	0.78	0.78	0.76	**0.007**	0.368	0.364	0.35	**0.0049**

of a total of 500 in the affected area, whereas HQGA-strong covers 290, a difference of just 10 mobile users. The same trend is observed with 20 UAVs. Regarding the standard deviation, with 5 UAVs, the HQGA-strong's deviation is 33% lower than that of HQGAFARA; with 10 UAVs, HQGAFARA's deviation is 35% lower than HQGA-strong; and with 20 UAVs, HQGA-strong's deviation is 30% lower than HQGAFARA. These results indicate that HQGA-strong is more robust than HQGAFARA, as it exhibits a lower standard deviation in two out of three tests.

The second set of experiments focused on evaluating the statistical performance of quantum and classical algorithms. The objective was to ascertain whether there are statistically significant differences in accuracy between the two types of algorithms. To achieve this, hypothesis tests were conducted utilizing data from both worst-case and best-case scenarios, across configurations of five, ten, and twenty UAVs. In each scenario, the experimental setup was as follows: **First Experiment:** Utilized five UAVs ($M = 5$). **Second Experiment:** Employed ten UAVs ($M = 10$). **Third Experiment:** Engaged twenty UAVs ($M = 20$). The results of these analyses are systematically presented across six experiments, detailed in Tables 5 and 6.

To determine the appropriate hypothesis test, either parametric or non-parametric, for each scenario, the following steps are followed:

1. Perform the Shapiro-Wilk test to determine if a sample is normally distributed. Based on this test, a decision is made to apply either a parametric or non-parametric test. A significance level of $\alpha = 0.01$ is used in all experiments, implying a 1% risk of incorrectly rejecting the null hypothesis H_0. The results are documented in Tables 5 and 6, where the results in each row indicate the p-value for the Shapiro-Wilk test and the t-value for the Welch test. The decision to use the Shapiro-Wilk test was based on the number of samples generated by each algorithm ($N = 30$).

Table 5 Statistical tests regarding accuracy in the best case scenario with 5, 10, and 20 UAVs. α represents the significance level value and M is the number of UAVs. The number of samples for both tests was 30

M	Shapiro-Wilk test (α = 0.01)			Welch t-student test (α = 0.001)		
	5	10	20	5	10	20
Algorithm						
GA-100	0.111	0.729	0.422	–	–	–
HQGAFARA_boots	0.9161	0.051	0.688	–13.427	–17.374	–25.748
HQGA-strong_boots	0.066	0.033	0.012	–15.782	–12.636	–18.601
Hypothesis H0						
GA-100	True	True	True	–	–	–
HQGAFARA_boots	True	True	True	False	False	False
HQGA-strong_boots	True	True	True	False	False	False

The Shapiro-Wilk test is ideal for detecting deviations from normality when the sample size is small to moderate, as in our case.

2. Depending on the Shapiro-Wilk test outcomes, one of the following approaches is taken:

   (a) If all results support H_0 (indicating normal distribution), apply the parametric Welch's t-test with $\alpha = 0.001$.
   (b) If all results reject H_0 (indicating non-normal distribution), apply the non-parametric Wilcoxon signed-rank test with $\alpha = 0.001$.
   (c) If results are mixed, some supporting and some rejecting H_0, employ bootstrapping methods to normalize distributions where necessary. Following normalization, the Welch's t-test is applied to these data. Results from this approach are identified in tables by a "boots" label next to the algorithm name.

The Table 5 documents the results of statistical tests assessing accuracy in best case scenarios with 5, 10, and 20 UAVs. The GA with a population of 100 chromosomes was selected for these tests due to its superior mean accuracy observed in the overall performance tests (Table 4), compared to its configurations with 32 and 1024 chromosomes.

In the best case scenario, the two quantum genetic algorithms exhibited non-normal distributions in their data. The GA showed a normal distribution, aligning with scenario 2c) where results from the Shapiro-Wilk test vary between true and false. Consequently, a bootstrapping method was applied to the data from the quantum genetic algorithms to achieve a normal distribution, as indicated by the "boots" label in the Table 5. This normalization allowed for the application of the Welch t-test to all algorithms. The results from this parametric test in Table 5 confirm that the null hypothesis H_0 is false for both quantum algorithms with 5, 10, and 20

Table 6 Statistical tests regarding accuracy in the worst case scenario with 5, 10, and 20 UAVs. α represents the significance level value and M is the number of UAVs. The number of samples for both tests was 30

M	Shapiro-Wilk test (α = 0.01)			Welch t-student test (α = 0.001)		
	5	10	20	5	10	20
Algorithm						
GA-100	0.344	0.519	0.125	–	–	–
HQGAFARA_boots	0.855	0.554	0.756	–13.083	–9.779	–18.638
HQGA-strong_boots	0.569	0.539	0.105	–16.291	–12.630	–17.112
Hypothesis H0						
GA-100	True	True	True	–	–	–
HQGAFARA_boots	True	True	True	False	False	False
HQGA-strong_boots	True	True	True	False	False	False

UAVs, indicating a significant accuracy advantage over the GA, demonstrating their supremacy in accuracy in the best case scenarios with 5, 10, and 20 UAVs.

The Table 6 presents the results of statistical tests regarding accuracy in the worst case scenario with 5, 10, and 20 UAVs. The Welch t-student test, a parametric method, was applied in this case, showing that the null hypothesis H0 is false with 5, 10, and 20 UAVs. This indicates a significant difference in accuracy between quantum genetic algorithms and GA. Specifically, the HQGAFARA and HQGA-strong demonstrated superior accuracy, as evidenced in Table 4, where both algorithms exhibited better accuracy values compared to GA. This evidence confirms the dominance of quantum genetic algorithms over GA in the worst case scenario for configurations involving 5, 10, and 20 UAVs. Additionally, the consistency of quantum algorithms outperforming GA in both best and worst scenarios underscores their robust capabilities in enhancing accuracy.

The third set of experiments was dedicated to analyzing the statistical performance of quantum and classical algorithms in terms of convergence time. In these analyses, we compared the number of generations required by each algorithm to find the optimal value. Using the same statistical testing procedure as the second set of experiments, we determined whether significant differences exist between classical and quantum algorithms regarding the speed of convergence.

The experiments were structured as follows for both best-case and worst-case scenarios: The first experiment involved five UAVs ($M = 5$). The second experiment used ten UAVs ($M = 10$). The third experiment employed twenty UAVs ($M = 20$). Results from these analyses are documented across Tables 7 and 8, which detail the convergence performance for each scenario and algorithm configuration.

Table 7 describes the performance of the GA with a population of 100, HQGA-FARA, and HQGA-strong regarding convergence time in the best case scenario. With 5 UAVs, the Shapiro-Wilk test demonstrated that all algorithms exhibit a non-normal distribution in their data. Consequently, the Wilcoxon signed-rank test was applied.

Table 7 Statistical tests regarding convergence time in the best case scenario with 5, 10, and 20 UAVs. α represents the significance level value and M is the number of UAVs. The number of samples for both tests was 30. For Welch and Wilcoxon test $\alpha = 0,001$; for Shapiro test $\alpha = 0,01$

M	Shapiro-Wilk test			Welch t-student test			Wilcoxon test		
	5	10	20	5	10	20	5	10	20
Algorithm									
GA-100	7.19E-04	0.241	1.72E-03	–	–	–	–	–	–
HQGAFARA_boots	–	11.6E-3	–	–	8.589	–	–	–	–
HQGAFARA	5.19E-10	–	5.06E-09	–	–	–	1	–	0
HQGA-strong_boots	–	0.2	–	–	6.683	–	–	–	–
HQGA-strong	2.95E-10	–	7.79E-09	–	–	–	83	–	3
Hypothesis H0									
GA-100	False	True	False	–	–	–	–	–	–
HQGAFARA_boots	–	True	–	–	False	–	–	–	–
HQGAFARA	False	–	False	–	–	–	False	–	False
HQGA-strong_boots	–	True	–	–	False	–	–	–	–
HQGA-strong	False	–	False	–	–	–	True	–	False

Table 8 Statistical tests regarding convergence time in the worst case scenario with 5, 10, and 20 UAVs. α represents the significance level value and M is the number of UAVs. The number of samples for both tests was 30. For Welch and Wilcoxon test $\alpha = 0,001$; for Shapiro test $\alpha = 0,01$

M	Shapiro-Wilk test			Welch t-student test			Wilcoxon test		
	5	10	20	5	10	20	5	10	20
Algorithm									
GA-100	0.059	0.208	8.7E-03	–	–	–	–	–	–
HQGAFARA_boots	0.156	0.098	–	5.341	8.799	–	–	–	–
HQGAFARA	–	–	2.38E-07	–	–	–	–	–	8
HQGA-strong_boots	0.12	0.114	–	4.315	9.025	–	–	–	–
HQGA-strong	–	–	9.78E-10	–	–	–	–	–	0
Hypothesis H0									
GA-100	True	True	False	–	–	–	–	–	–
HQGAFARA_boots	True	True	–	False	False	–	–	–	–
HQGAFARA	–	–	False	–	–	–	–	–	False
HQGA-strong_boots	True	True	–	False	False	–	–	–	–
HQGA-strong	–	–	False	–	–	–	–	–	False

The results indicated no significant difference between HQGA-strong and the GA concerning convergence time. However, for HQGAFARA, the null hypothesis H0 is rejected, indicating a significant difference from the GA. In the experiments, HQGA-FARA requires approximately 18 generations to find the optimal value, while the GA needs about 180 generations. This demonstrates that HQGAFARA is approximately ten times faster than GA at locating the optimal solution with 5 UAVs in the best case scenario. With 10 UAVs, both quantum genetic algorithms, HQGAFARA and

HQGA-strong, exhibited non-normal distributions in their data, while the GA displayed a normal distribution. To facilitate a parametric analysis, the bootstrapping method was applied to normalize the data of HQGAFARA and HQGA-strong. Subsequently, the Welch t-test was conducted to assess the significance of differences between the classical and quantum algorithms. The null hypothesis H0 was rejected for both quantum genetic algorithms, signifying a significant difference in convergence time compared to the GA. In the experiments, HQGAFARA achieved the best solution in just the first generation, while HQGA-strong required 74 generations. In contrast, the GA took approximately 345 generations to find the optimal solution. These results confirm that HQGA-strong is about four times faster than the GA in finding the optimal value in the best case scenario with 10 UAVs. With 20 UAVs, the three algorithms-HQGAFARA, HQGA-strong, and the GA-exhibited non-normal distributions in their data. Consequently, a non-parametric test was appropriate for analyzing the data. The Wilcoxon signed-ranks test was applied, leading to the rejection of the null hypothesis H0 for both quantum genetic algorithms. According to the experiments the results favor the quantum genetic algorithms: HQGAFARA achieved the optimal solution in approximately 68 generations, while HQGA-strong required about 109 generations to find the optimal solution. These outcomes underscore the efficiency of quantum genetic algorithms in rapidly converging to optimal solutions in scenarios involving 20 UAVs.

The Table 8 present statistical results regarding convergence time in the worst case scenario with 5, 10, and 20 UAVs, respectively. With 5 and 10 UAVs, the bootstrapping method was utilized to normalize the data for HQGAFARA and HQGA-strong due to their non-normal distribution. In contrast, with 20 UAVs reports that the Shapiro-Wilk test confirmed a non-normal distribution for the GA, HQGAFARA, and HQGA-strong, prompting the application of the non-parametric Wilcoxon signed ranks test. The null hypothesis H0 was rejected in all cases with 5, 10, and 20 UAVs, indicating a significant difference in convergence time between the GA and the quantum genetic algorithms in the worst case scenario. In the experiments, the quantum genetic algorithms consistently outperformed the GA across all tests. Particularly in scenarios with 10 and 20 UAVs, HQGA-strong required fewer generations to find the optimal solution compared to HQGAFARA. This efficiency advantage is attributed to HQGA-strong's less complex quantum circuitry used in its recombination process. The complexity of a quantum circuit, typically measured by the number of quantum operations (length or depth), plays a crucial role in its efficiency [41]. Shorter circuits are generally more efficient or less complex. As the number of UAVs (qubits) increases, the quantum circuit must perform more operations, yet HQGA-strong maintains greater efficiency than HQGAFARA with 10 and 20 UAVs in the worst case scenario.

5 Conclusion

A Hybrid Quantum Genetic Algorithm with a strongly entanglement recombination process (HQGA-strong) to maximize the coverage area for a set of unmanned aerial vehicles in a natural disaster location is presented in this paper. Three sets of experiments were conducted to evaluate the overall performance of the quantum genetic algorithm regarding accuracy and convergence time. Additionally, a hybrid quantum genetic algorithm with a fuzzy adaptive rotation angle and a classical genetic algorithm were developed to compare with our HQGA-strong. Two scenarios were used to evaluate the performance of the quantum and classical algorithms, the best case and worst case scenario. For each scenario, three different numbers of UAVs were employed.

The contributions of this paper to the currently state-of-the-art literature are the following:

- The integration of a strongly entanglement quantum circuit as a recombination operator. This quantum circuit demonstrated based on statistical results its superiority regarding convergence time compared with the Deutsch-Jozsa quantum circuit implemented in HQGAFARA.
- The implementation of a hybrid quantum genetic algorithm with a strongly entanglement quantum circuit (HQGA-strong) for maximizing the coverage area for a set of UAVs in a disaster location. The proposed HQGA-strong demonstrated better accuracy results compared with the GA and its counterpart HQGAFARA in the worst case scenario, where the number of transmission channels tests the performance of classical and quantum algorithms.

The experimental results regarding overall performance and the statistical tests about accuracy and convergence time demonstrated the supremacy of both quantum genetic algorithms, HQGAFARA and HQGA-strong compared to GA. In the experimental results regarding overall performance (see Table 4), six results about mean solutions were presented, three with the best case scenario and three with the worst case scenario; each set with 5, 10, and 20 UAVs. Comparing only the results of HQGAFARA and HQGA-strong, the HQGAFARA obtained better accuracy in three of the six results, best case with 10 and 20 UAVs, and worst case with 20 UAVs; in average the HQGAFARA obtained 1.4% better results in the mean solutions compared with the HQGA-strong. The HQGA-strong showed better mean solutions in the best case scenario with 5 UAVs and in the worst case scenario with 5 and 10 UAVs; in average the HQGA-strong obtained 0.6% better results in the mean solutions compared with the HQGAFARA. The difference between the average results of HQGAFARA (1.4%) and HQGA-strong (0.6%) is less than 1%. Therefore, we can say that HQGA-strong performs similar to HQGAFARA in the problem of maximizing the coverage area of a set of UAVs in a natural disaster location. From Tables 7 and 8, we consider the number of generations that each algorithm needs to find the optimal solution. The results presented in the statistical results from Tables 7 and 8 showed that in most of the cases there is a significant difference between classical

and quantum algorithms. According to the experiments, the difference is in favor of quantum algorithms since they obtain less generations to reach the optimal value compared to GA. The HQGAFARA found the best solutions in less generations in four of the six we consider the results, the best case with 5, 10, and 20 UAVs; and the worst case with 5 UAVs. Whereas, the HQGA-strong found the best solution in less generations in the worst case with 10 and 20 UAVs. This scenario is the most difficult since the number of transmission channels is less than the best case scenario, challenging the performance of the algorithms; here, the HQGA-strong demonstrated a better convergence time finding the best solutions in less generations than HQGAFARA.

All the experiments were carried out on the IBM Qiskit Platform using its Qiskit library. The GA, HQGAFARA and HQGA-strong were developed in Python. One of the main limitations of this work is the time it takes for each algorithm to run on a real quantum computer, since not all the time there was a quantum computer available to run the experiments. For that reason we chose to perform all the experiments using a simulator. Another kind of limitations that currently affect the performance of quantum computers are the error rates, since the qubits are intrinsically susceptible to decoherence. This intrinsic decoherence occurs due to the fragile nature of qubits, which can be easily disturbed by external factors such as temperature, or electromagnetic fields.

For future work, we aim to evaluate our quantum proposal in more expansive geographical areas while simultaneously increasing the number of UAVs (qubits) to expand the scope of our coverage. Also, we will implement our HQGA-strong in more challenging conditions by modifying the parameters used in the emergency LTE mobile network. In addition, the experiments will be implemented on a real quantum computer to test the efficiency of quantum algorithms in the presence of noise and decoherence. Finally, we will refine our hybrid quantum algorithm in order to determine the optimal path between the UAV and the previously identified strategic position. The process involves the integration of path planning and strategic positioning in disaster areas. It will be necessary to combine synergistic and multidisciplinary efforts from different fields for the effective design of the route generated by our HQGA-strong, as the kinematic model and constraints of the UAVs must be taken into account to ensure they follow the predetermined route.

Acknowledgements We thank Instituto Politécnico Nacional (IPN), the Comisión de Fomento y Apoyo Académico del IPN (COFAA), and the Mexican National Council on Humanities, Science and Technology (CONAHCYT) for supporting our research activities.

Declarations

• Funding. This research was funded by the Instituto Politécnico Nacional under Grant Number SIP20230104. We also thank the Comisión de Fomento y Apoyo Académico del IPN (COFAA), and the Mexican National Council of Humanity Science and Technology (CONAHCYT) for supporting our research activities.

• Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use). The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

- Ethics approval and consent to participate. Not applicable.
- Consent for publication. Not applicable.
- Data availability. Not applicable.
- Materials availability. Not applicable.
- Code availability. Not applicable.
- Author contribution. Conceptualization, Enrique Ballinas, Oscar Montiel, Anabel Martínez-Vargas; Formal analysis, Enrique Ballinas; Funding acquisition, Oscar Montiel; Investigation, Enrique Ballinas, Oscar Montiel, Anabel Martínez-Vargas; Methodology, Enrique Ballinas, Oscar Montiel; Project administration, Oscar Montiel, Anabel Martínez-Vargas; Resources, Oscar Montiel, Anabel Martínez-Vargas; Software, Enrique Ballinas; Supervision, Oscar Montiel, Anabel Martínez-Vargas; Validation, Oscar Montiel, Anabel Martínez-Vargas; Visualization, Enrique Ballinas; Writing—original draft, Enrique Ballinas; Writing—review and editing, Oscar Montiel, Anabel Martínez-Vargas.

References

1. Wang, Q., Li, W., Yu, Z., Abbasi, Q., Imran, M., Ansari, S., Sambo, Y., Wu, L., Li, Q., Zhu, T.: An overview of emergency communication networks. Remote Sensing **15**(6) (2023). https://doi.org/10.3390/rs15061595
2. Cangialosi, J.P., Kimberlain, T.B.: Hurricane odile (ep152014). National Hurricane Center Tropical Cyclone Report (2015)
3. Yabe, T., Ukkusuri, S.V., C. Rao, P.S.: Mobile phone data reveals the importance of pre-disaster inter-city social ties for recovery after hurricane maria. Applied Network Science **4**(1), 98 (2019). https://doi.org/10.1007/s41109-019-0221-5
4. Feng, W.: Summary of global earthquake disasters in 2018 (2019)
5. Shugar, D.H., Jacquemart, M., Shean, D., Bhushan, S., Upadhyay, K., Sattar, A., Schwanghart, W., McBride, S., Vries, M.V.W., Mergili, M., Emmer, A., Deschamps-Berger, C., McDonnell, M., Bhambri, R., Allen, S., Berthier, E., Carrivick, J.L., Clague, J.J., Dokukin, M., Dunning, S.A., Frey, H., Gascoin, S., Haritashya, U.K., Huggel, C., Kääb, A., Kargel, J.S., Kavanaugh, J.L., Lacroix, P., Petley, D., Rupper, S., Azam, M.F., Cook, S.J., Dimri, A.P., Eriksson, M., Farinotti, D., Fiddes, J., Gnyawali, K.R., Harrison, S., Jha, M., Koppes, M., Kumar, A., Leinss, S., Majeed, U., Mal, S., Muhuri, A., Noetzli, J., Paul, F., Rashid, I., Sain, K., Steiner, J., Ugalde, F., Watson, C.S., Westoby, M.J.: A massive rock and ice avalanche caused the 2021 disaster at chamoli, indian himalaya. Science **373**(6552), 300–306 (2021). https://doi.org/10.1126/science.abh4455. https://www.science.org/doi/pdf/10.1126/science.abh4455
6. Ortiz, G., Aznar-Crespo, P., Aledo, A.: Developing and pilot-testing warning messages for risk communication in natural disasters. Environment Systems and Decisions **44**(2), 239–258 (2024). https://doi.org/10.1007/s10669-023-09924-z
7. Alenazi, M.J.F.: Enrn: A system for evaluating network resilience against natural disasters. Mathematics **11**(20) (2023). https://doi.org/10.3390/math11204250
8. Sambo, Y.A., Klaine, P.V., Nadas, J.P.B., Imran, M.A.: Energy minimization uav trajectory design for delay-tolerant emergency communication. In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6 (2019). https://doi.org/10.1109/ICCW.2019.8757127
9. Huang, D., Cui, M., Zhang, G., Chu, X., Lin, F.: Trajectory optimization and resource allocation for uav base stations under in-band backhaul constraint. EURASIP Journal on Wireless Communications and Networking **2020**(1), 83 (2020). https://doi.org/10.1186/s13638-020-01700-w
10. Zheng, Y.-J., Chen, S.-Y., Ling, H.-F.: Evolutionary optimization for disaster relief operations: A survey. Applied Soft Computing **27**, 553–566 (2015). https://doi.org/10.1016/j.asoc.2014.09.041

11. Talbi, E.G.: Metaheuristics: From Design to Implementation. Wiley Series on Parallel and Distributed Computing. Wiley, ??? (2009). https://books.google.com.mx/books?id=SIsa6zi5XV8C
12. Zhu, J., Guo, K., He, P., Huang, H., Malekian, R., Sun, Y.: An effective trajectory planning heuristics for uav-assisted vessel monitoring system. Peer-to-Peer Networking and Applications **17**(4), 2491–2506 (2024). https://doi.org/10.1007/s12083-024-01730-x
13. Jain, S., Bharti, K.K.: A combinatorial optimization model for post-disaster emergency resource allocation using meta-heuristics. Soft Computing **27**(18), 13595–13611 (2023). https://doi.org/10.1007/s00500-022-07176-8
14. Hooshyar, M., Huang, Y.-M.: Meta-heuristic algorithms in uav path planning optimization: A systematic review (2018–2022). Drones **7**(12) (2023). https://doi.org/10.3390/drones7120687
15. Abid, M., El Kafhali, S., Amzil, A., Hanini, M.: Optimization of uav flight paths in multi-uav networks for efficient data collection. Arabian Journal for Science and Engineering (2024). https://doi.org/10.1007/s13369-024-09369-9
16. Yahia, H.S., Mohammed, A.S.: Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: a systematic review. Environmental Monitoring and Assessment **195**(1), 30 (2022). https://doi.org/10.1007/s10661-022-10590-y
17. Ozkan, O.: Optimization of the distance-constrained multi-based multi-uav routing problem with simulated annealing and local search-based matheuristic to detect forest fires: The case of turkey. Applied Soft Computing **113**, 108015 (2021). https://doi.org/10.1016/j.asoc.2021.108015
18. Aydin, N., Yilmaz, O., Deveci, M., Lv, Z.: Heuristics based optimization for multidepot drone location and routing problem to detect post-earthquake damages. IEEE Transactions on Intelligent Transportation Systems **25**(1), 850–858 (2024). https://doi.org/10.1109/TITS.2022.3190698
19. Rodríguez-Cortés, G., Martínez-Vargas, A., Montiel Ross, O.: The Deployment of Unmanned Aerial Vehicles at Optimized Locations to Provide Communication in Natural Disasters, pp. 501–513 (2018). https://doi.org/10.1007/978-3-319-71008-2_36
20. Gupta, M., Varma, S.: Metaheuristic-based optimal 3d positioning of uavs forming aerial mesh network to provide emergency communication services. IET Communications **15**, 1297–1314 (2021). https://doi.org/10.1049/cmu2.12112
21. Fernandez, S.A., Carvalho, M.M., Silva, D.G.: A hybrid metaheuristic algorithm for the efficient placement of uavs. Algorithms **13**(12) (2020). https://doi.org/10.3390/a13120323
22. Akram, T., Awais, M., Naqvi, R., Ahmed, A., Naeem, M.: Multicriteria uav base stations placement for disaster management. IEEE Systems Journal **14**(3), 3475–3482 (2020). https://doi.org/10.1109/JSYST.2020.2970157
23. Rottondi, C., Malandrino, F., Bianco, A., Chiasserini, C.F., Stavrakakis, I.: Scheduling of emergency tasks for multiservice uavs in post-disaster scenarios. Computer Networks **184**, 107644 (2021). https://doi.org/10.1016/j.comnet.2020.107644
24. Wang, Y., Wang, W.: Quantum-inspired differential evolution with grey wolf optimizer for 0-1 knapsack problem. Mathematics **9**(11) (2021). https://doi.org/10.3390/math9111233
25. Huang, Z., Zhang, J., Wu, M., Li, X., Dong, Y.: Evolutionary method of sink node path planning guided by the hamiltonian of quantum annealing algorithm. IEEE Access **9**, 53466–53479 (2021). https://doi.org/10.1109/ACCESS.2021.3070650
26. Ballinas, E., Montiel, O., Martínez-Vargas, A., Rodríguez-Cortés, G.: Hybrid quantum genetic algorithm with fuzzy adaptive rotation angle for efficient placement of unmanned aerial vehicles in natural disaster areas. Axioms **13**(1) (2024). https://doi.org/10.3390/axioms13010048
27. Zhang, X., Xia, S., Li, X.: Quantum behavior-based enhanced fruit fly optimization algorithm with application to uav path planning. INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE SYSTEMS **13**(1), 1315–1331 (2020). https://doi.org/10.2991/ijcis.d.200825.00
28. Hu, C., Xia, Y., Zhang, J.: Adaptive operator quantum-behaved pigeon-inspired optimization algorithm with application to uav path planning. ALGORITHMS **12**(1) (2019). https://doi.org/10.3390/a1201000

29. Li, S., Deng, Y.: Quantum-entanglement pigeon-inspired optimization for unmanned aerial vehicle path planning. AIRCRAFT ENGINEERING AND AEROSPACE TECHNOLOGY **91**(1), 171–181 (2019). https://doi.org/10.1108/AEAT-03-2018-010
30. Mousavi, S., Afghah, F., Ashdown, J.D., Turck, K.: Use of a quantum genetic algorithm for coalition formation in large-scale uav networks. AD HOC NETWORKS **87**, 26–36 (2019). https://doi.org/10.1016/j.adhoc.2018.11.00
31. Maity, R., Mishra, R., Pattnaik, P.K.: Analysis of path finding techniques for flying robots through intelligent decision-making algorithms in quantum inspired computing environment. Wireless Personal Communications **135**(3), 1561–1580 (2024). https://doi.org/10.1007/s11277-024-11125-z
32. Kumar, A., Bhatia, S., Kaushik, K., Gandhi, S.M., Devi, S.G., De J. Pacheco, D.A., Mashat, A.: Survey of promising technologies for quantum drones and networks. IEEE Access **9**, 125868–125911 (2021). https://doi.org/10.1109/ACCESS.2021.3109816
33. Rizk, Y., Awad, M.: A quantum genetic algorithm for pickup and delivery problems with coalition formation. Procedia Computer Science **159**, 261–270 (2019). https://doi.org/10.1016/j.procs.2019.09.181. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019
34. Pustokhina, I.V., Pustokhin, D.A., Lydia, E.L., Elhoseny, M., Shankar, K.: Energy efficient neuro-fuzzy cluster based topology construction with metaheuristic route planning algorithm for unmanned aerial vehicles. Computer Networks **196**, 108214 (2021). https://doi.org/10.1016/j.comnet.2021.108214
35. Alekhya, V., Reddy, N.V.U., Singh, J., Boddu, B., Sobti, R., Hameed, A.A.: High-dimensional data processing using quantum-inspired evolutionary algorithms for homeland security imaging systems. In: 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), pp. 1184–1189 (2024). https://doi.org/10.1109/IC3SE62002.2024.10592971
36. Thomas, A., Mannone, M., Seidita, V., Chella, A.: Swarm robotics and quantum computing in simulated environments (2024)
37. Hidary, J.D.: Quantum Computing: An Applied Approach, (2021). https://books.google.com.mx/books?id=T7VFEAAAQBAJ
38. Zhong, H.-S., Deng, Y.-H., Qin, J., Wang, H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Wu, D., Gong, S.-Q., Su, H., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Renema, J.J., Lu, C.-Y., Pan, J.-W.: Phase-programmable gaussian boson sampling using stimulated squeezed light. Phys. Rev. Lett. **127**, 180502 (2021). https://doi.org/10.1103/PhysRevLett.127.180502
39. Born, M.: Zur quantenmechanik der stoßvorgänge. Zeitschrift für Physik **37**(12), 863–867 (1926). https://doi.org/10.1007/BF01397477
40. Yanofsky, N., Manucci, M.: Quantum Computing for Computer Scientists, (2008)
41. Williams, C.P.: Explorations in Quantum Computing, (2011)
42. Rodríguez-Cortés, G.L., Martínez-Vargas, A., Montiel-Ross, O.H., Cosío-León, M., Martínez, D.M.: Deployment of unmanned aerial vehicles for maximum coverage in emergency scenarios using the (1+1) evolution strategy with one-fifth success rule. In: 2021 Mexican International Conference on Computer Science (ENC), pp. 1–7 (2021). https://doi.org/10.1109/ENC53357.2021.9534767
43. Martínez-Vargas, A., Rodríguez-Cortés, G.L., Montiel-Ross, O.: Comparative representations of a genetic algorithm to locate unmanned aerial vehicles in disaster zones. Engineering Letters **27**(2) (2019)
44. Martínez-Vargas, A., Andrade, Á.G.: Deployment analysis and optimization of heterogeneous networks under the spectrum underlay strategy. EURASIP Journal on Wireless Communications and Networking **2015**(1), 55 (2015). https://doi.org/10.1186/s13638-015-0276-7
45. Wittek, P.: 3 - quantum mechanics. In: Wittek, P. (ed.) Quantum Machine Learning, pp. 25–39. Academic Press, Boston (2014). https://doi.org/10.1016/B978-0-12-800953-6.00003-7. https://www.sciencedirect.com/science/article/pii/B9780128009536000037

46. Rubio, Y., Olvera, C., Montiel, O.: Quantum-inspired evolutionary algorithms on ibm quantum experience. Engineering Letters **29**(4), 1573–1584 (2021). Publisher Copyright: 2021, International Association of Engineers. All rights reserved.
47. Haupt, R., Haupt, S.-E.: Practical Genetic Algorithms, (2004)
48. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Physical Review A **101**(3) (2020). https://doi.org/10.1103/physreva.101.032308

Review on Quantum Optimization of Type-1, Type-2 and Type-3 Fuzzy Systems

Oscar Castillo

Abstract In this paper we are presenting an overview of the existing works that have been published in the quantum optimization of type-1, type-2 and type-3 fuzzy systems, considering both theoretical contributions and applications. The analysis was done considering all kinds of publications: journal papers, chapters in books and conference papers. The database was obtained from Scopus system of Elsevier, which contains most of the indexed publications in the world in all areas. It is relevant to consider Quantum Optimization when designing higher types of fuzzy systems as the number of parameter is higher and correspondingly the design complexity is also higher. Based on the obtained data, relevant conclusions were obtained about the development of the quantum optimization of fuzzy systems, as well as future trends.

Keywords Quantum optimization · Type-2 fuzzy systems · Type-3 fuzzy systems

1 Introduction

Fuzzy logic since its inception [1] has been successfully applied in a plethora of areas [2, 3]. The main idea of fuzzy logic was to model linguistic uncertainty in real human decision-making, and the original fuzzy logic is now called type-1 as it was the initial version of this idea. More recently, type-2 was proposed and developed to offer a better model of uncertainty [4] and consequently more complex problems were solved [5, 6]. Still, more complex problems can be tackled with a most recent version of fuzzy theory, called type-3 fuzzy logic [7–9], which offers an even better model of uncertainty a now some recent successful applications have been achieved, like in [10–17].

These previous works consider problems from different perspectives, but in general the decision-making process is viewed as a problem under uncertainty. For

O. Castillo (✉)
Tijuana Institute of Technology, TecNM, Tijuana, BC, Mexico
e-mail: ocastillo@tectijuana.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_12

this reason, we decided to consider analyzing the existing works utilizing type-3 fuzzy logic, as this approach offers in theory a more powerful way to represent uncertainty. Accordingly, the contribution of this work is the study and analysis of type-3 fuzzy literature. We can mention that there is currently some recent success in handling prediction with type-3, like in [18–20]. Also, there are some good results in control with type-3, like in [21–23]. In addition, the application of type-3 fuzzy logic in enhancing metaheuristics, like in [24]. The aim of type-3 in all cases is to capture in a better way the involved uncertainty in decision-making. The main problem in applying fuzzy systems is the finding their optimal structure and parameters. This is where optimization techniques can help in obtaining better fuzzy systems for the applications. The problem is even more complex for higher types of fuzzy systems, like type-2 and type-3, and this is where we believe quantum optimization could help in enhancing the results. Quantum optimization algorithms, as in [25, 26], can be applied to optimally design the fuzzy systems.

The rest of the paper is as follows. In Sect. 2 we offer some basic concepts of type-1, type-2 and type-3 fuzzy logic. Section 3 describes a review of the existing papers on quantum optimization of fuzzy systems. Section 4 shows the analysis of results, and Sect. 5 offers the conclusions.

2 From Type-1 to Type-3 Fuzzy Sets

There exists different types of Fuzzy Sets, in this case, for example: Type-1 [1–3], Type-2 [27, 28] and Type-3 [7]. All of them with different capabilities for handling uncertainty and with different computational complexities. The definitions of Type-1, Type-2, and Type-3 fuzzy sets are expressed in (1)–(3), respectively.

$$A^{(1)} = \{(x, \mu_A(x)) | \forall x \in [0, 1]\} \tag{1}$$

$$A^{(2)} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u)\right) \middle| \forall u \in J_x \subseteq [0, 1] \right\} \tag{2}$$

$$A^{(3)} = \left\{ \left((x, u), \mu_{A^{(3)}}(x, u, v)\right) \middle| x \in X,\ u \in U \subseteq [0, 1],\ v \in V \subseteq [0, 1] \right\} \tag{3}$$

As can be noted, Type-2 Fuzzy Sets (T2 FSs) and Type-3 Fuzzy Sets (T3 FSs) represent implementation challenges because these systems require a greater computational cost when compared with respect to Type-1 Fuzzy Sets (T1 FSs). Since a T2 FS can be understood as a collection of embedded Type-1 membership functions (T1 MFs) over the secondary axis u and Type-3 Fuzzy Sets are a collection of embedded Type-2 membership functions (T2 MFs) over the third axis v, it is clear that a higher computer overhead is required to compute them. Equations (1)–(3) illustrate the general models of the presented type of Fuzzy Sets. However, there exist simplified approaches of these models to handle different levels of uncertainty, such as Interval

Type-2 Fuzzy Sets (IT2 FSs as a simplification of T2 FSs) and Interval Type-3 Fuzzy Sets (IT3 FSs as a simplification of T3 FSs).

3 Review on Quantum Optimization of Fuzzy Systems

In this section we describe the summary of the search results in Scopus when we use the query "quantum optimization of fuzzy systems", which produces 372 documents. Of this total, there are 159 journal papers, 107 review papers, 87 conference papers, 7 chapters in books, and one book. In Fig. 1 we illustrate the number of publications per year in this area. Table 1 shows the same information for the most recent year. Figure 2 illustrates the most prominent authors and Table 2 lists their number of publications. Figure 3 shows the documents produced by the institutions, where Jiangnan University and North China Electric Power University are the top institutions in this area.

Figure 4 illustrates the most productive countries in the area of quantum optimization of fuzzy systems. Table 3 list the respective countries with the most publications in this area.

Figure 5 illustrates the distribution of published documents according to their type with article in journal being the most prominent (42.7% of the total). Table 4 lists the corresponding information about the publications.

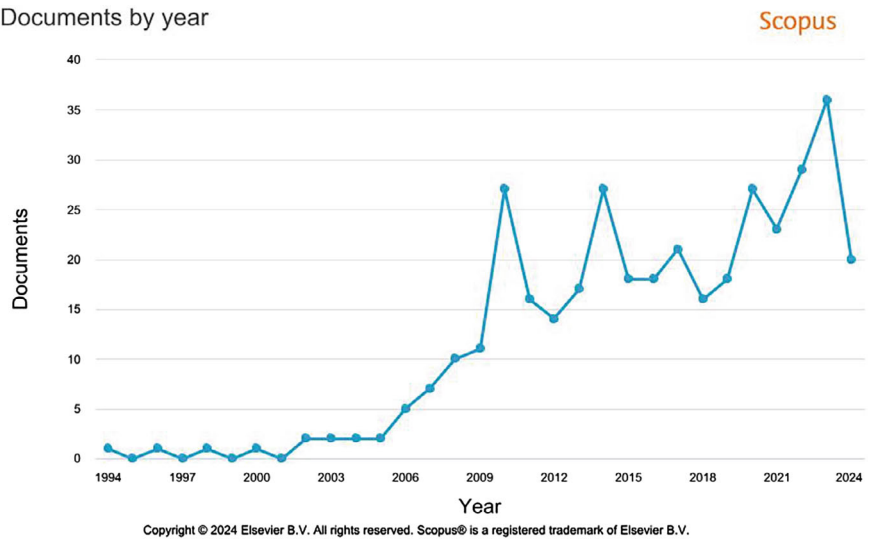


Fig. 1 Publications per year on quantum optimization of fuzzy systems

Table 1 Publications per year in quantum optimization of fuzzy systems

Year	Documents
2024	20
2023	36
2022	29
2021	23
2020	27
2019	18
2018	16
2017	21
2016	18

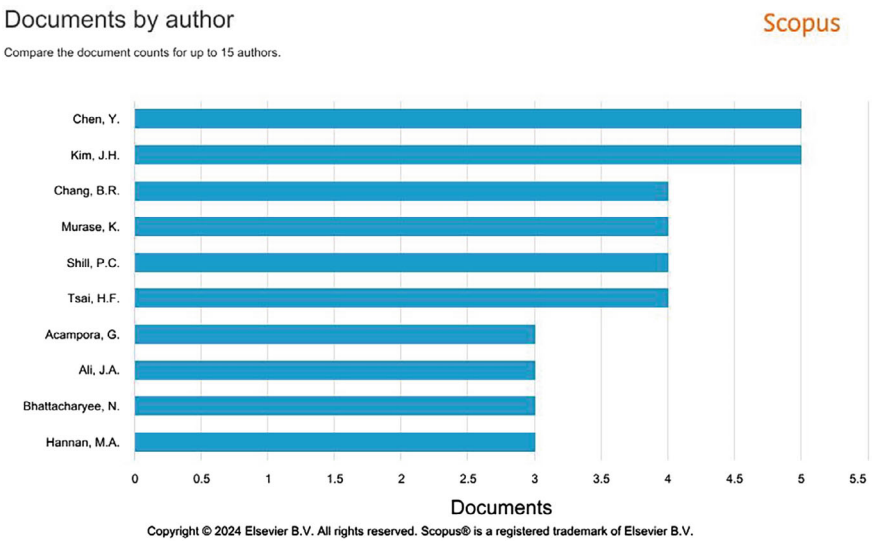


Fig. 2 Most prominent authors on quantum optimization of fuzzy systems

Table 2 Authors in quantum optimization of fuzzy systems

Author	Documents
Chen, Y.	5
Kim, J.H.	5
Chang., B.R.	4
Murase, K.	4
Shill, P.C.	4
Tsai, H.F.	4
Acampora, G.	3
Ali, J.A.	3
Bhattacharyee, N.	3

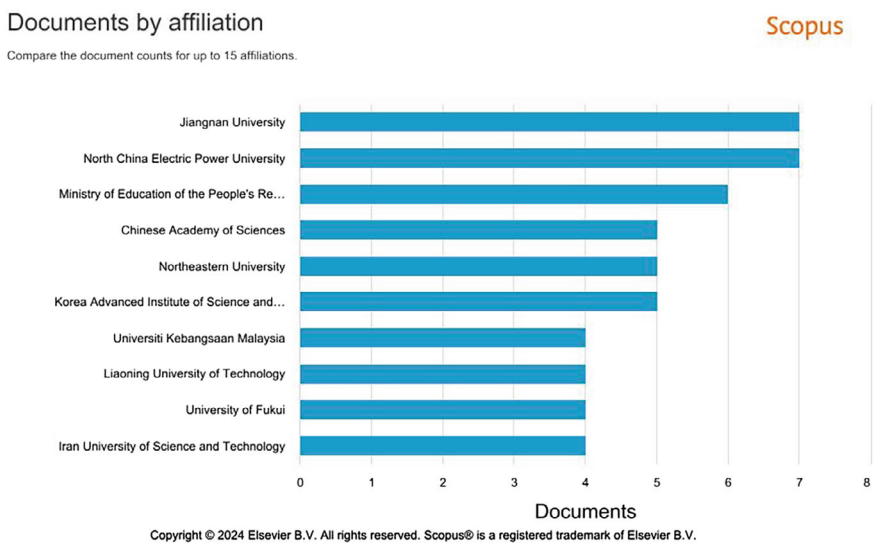

Documents by affiliation
Scopus
Compare the document counts for up to 15 affiliations.
Jiangnan University
North China Electric Power University
Ministry of Education of the People's Re…
Chinese Academy of Sciences
Northeastern University
Korea Advanced Institute of Science and…
Universiti Kebangsaan Malaysia
Liaoning University of Technology
University of Fukui
Iran University of Science and Technology
0 1 2 3 4 5 6 7 8
Documents
Copyright © 2024 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Fig. 3 Top institutions on quantum optimization of fuzzy systems

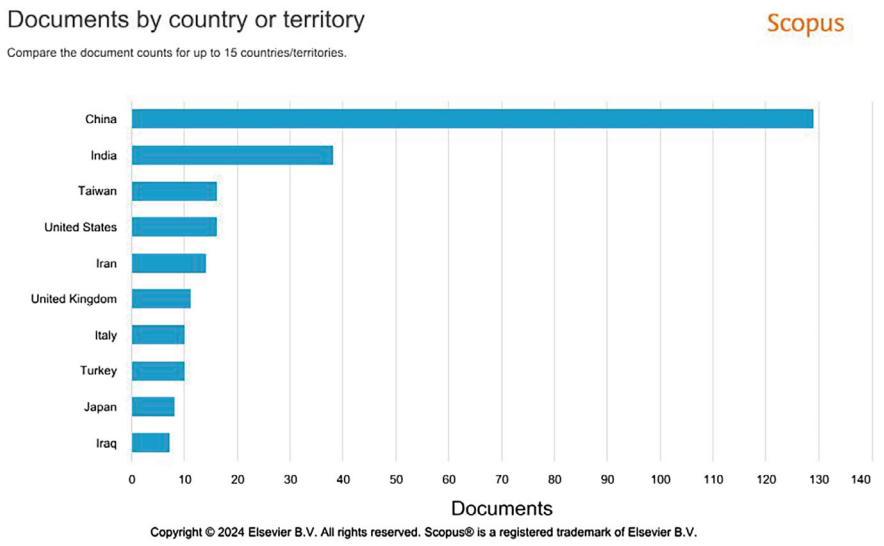


Fig. 4 Most productive countries in the area

Figure 6 exhibits the distribution of papers by subject area, in which the most prominent area is computer science, followed by engineering and mathematics. Table 5 summarizes the publications according to their respective area.

Table 3 Most productive countries in quantum optimization of fuzzy systems

Country	Documents
China	129
India	38
Taiwan	16
United States	16
Iran	14
United Kingdom	11
Italy	10
Turkey	10
Japan	8
Iraq	7
South Korea	7
Saudi Arabia	6
Australia	5
Brazil	5
Canada	5
Romania	5

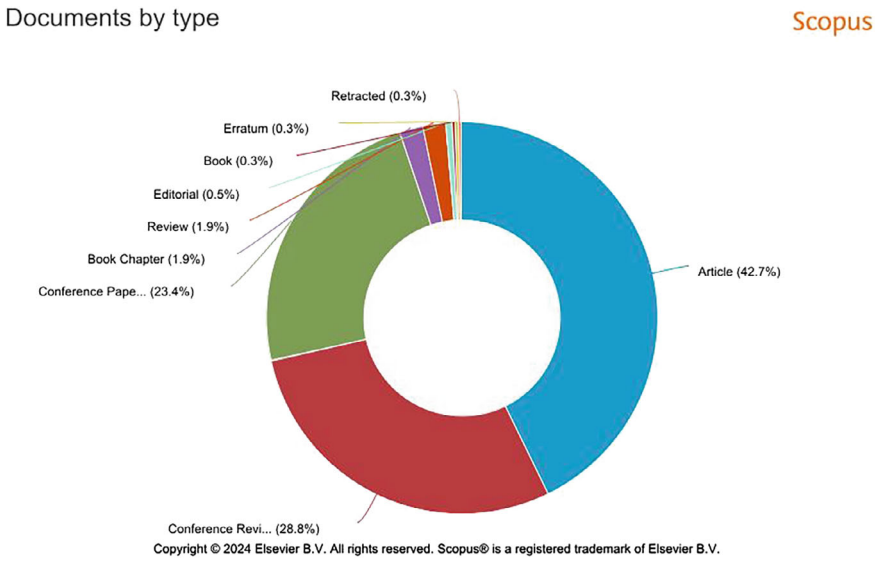


Fig. 5 Distribution of published documents according to their type

Table 4 Distribution of publications in quantum optimization of fuzzy systems

Author	Documents
Article	159
Conference review	107
Conference paper	87
Book chapter	7
Review	7
Editorial	2
Book	1
Erratum	1
Retracted	1

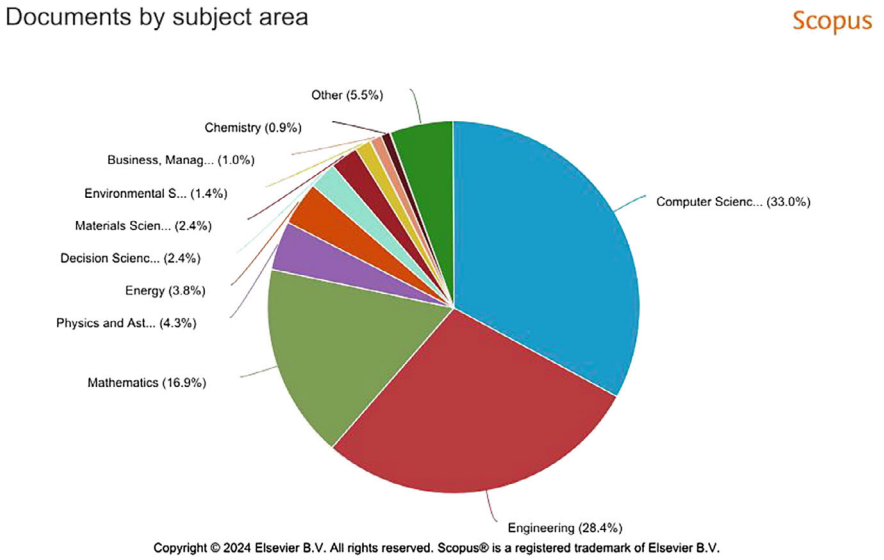


Fig. 6 Distribution of published documents according to their subject area

Table 5 Distribution of publications in quantum optimization of fuzzy systems according to the area

Area	Documents
Computer science	232
Engineering	200
Mathematics	119
Physics and astronomy	30
Energy	27
Decision sciences	17
Material sciences	17
Environmental science	10
Business, management and accounting	7

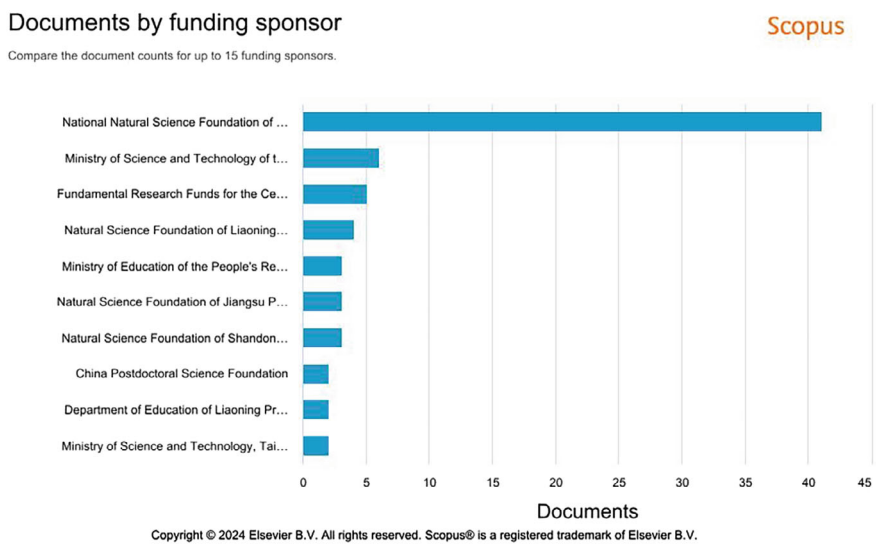

Documents by funding sponsor
Compare the document counts for up to 15 funding sponsors.
Scopus
National Natural Science Foundation of ...
Ministry of Science and Technology of t...
Fundamental Research Funds for the Ce...
Natural Science Foundation of Liaoning...
Ministry of Education of the People's Re...
Natural Science Foundation of Jiangsu P...
Natural Science Foundation of Shandon...
China Postdoctoral Science Foundation
Department of Education of Liaoning Pr...
Ministry of Science and Technology, Tai...
0
5
10
15
20
25
30
35
40
45
Documents
Copyright © 2024 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Fig. 7 Distribution of published documents according to their funding sponsor

Finally, Fig. 7 illustrates the top funding sponsors for quantum optimization of fuzzy systems. The top sponsor is National Natural Science Foundation of China.

4 Analysis and Future Trends for Quantum Optimization of Fuzzy Systems

In this section we describe the analysis of the data obtained from Scopus and also, we envision possible future trends for the area of quantum optimization of fuzzy systems. Based on the publications that have been made, we can say that most of the papers on quantum optimization of fuzzy systems have been applied in the computer science and engineering areas. We did notice that at the moment only fuzzy systems of type-1 and type-2 have been optimized with quantum algorithms, but we consider that type-3 will be more prominent in the future, as type-3 fuzzy systems have more parameters and are more complex. In addition, another topic that has not been explored is the use of fuzzy systems for achieving dynamic parameter adaptation in quantum algorithm, which could help in achieving even better results in many complex problems. Finally, there are also theoretical work on type-n fuzzy logic, which could also allow testing with quantum optimization on type-n fuzzy systems in the near future.

5 Conclusions

In this paper we have outlined a review of the publications on quantum optimization of fuzzy systems in recent years, which include papers in journals, conference proceeding and in edited books. The statistics of publications, authors and institutions were presented, as well as the analysis of the data and future trends for the area of quantum optimization of fuzzy systems. Based on the results and analysis we believe that this area is becoming a very important topic, and we expect that in the future more theoretical advances will come, as well as new applications will be developed. Future work in this area could include the optimization of type-3 and type-n fuzzy systems, as we believe that results could be enhanced with respect to traditional optimization.

References

1. Zadeh L.A. (1965) Fuzzy sets. Inf Control 8:338–353.
2. Zadeh, L. A., Knowledge representation in Fuzzy Logic, IEEE Transactions on knowledge data engineering, Vol. 1, p.89, 1989.
3. Zadeh, L. A., The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* **1975**, *8*, 199-249.
4. Mendel JM, Bob John RI (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127.
5. Ontiveros, E.; Melin, P.; Castillo, O. Comparative study of interval Type-2 and general Type-2 fuzzy systems in medical diagnosis. *Inf. Sci.* **2020**, *525*, 37–53.
6. Ontiveros-Robles, E.; Castillo, O.; Melin, P. Towards asymmetric uncertainty modeling in designing General Type-2 Fuzzy classifiers for medical diagnosis. *Expert Syst. Appl.* **2021**, *183*, 115370.
7. Castillo, O., Castro, J.R., Melin, P., Interval Type-3 Fuzzy Systems: Theory and Design, Springer, Cham, Switzerland, 2022.
8. Qasem S.N., Ahmadian A., Mohammadzadeh A., Rathinasamy S., Pahlevanzadeh B. A type-3 logic fuzzy system: Optimized by a correntropy based Kalman filter with adaptive fuzzy kernel size, Inform. Sci., 572 (2021), pp. 424-443
9. Mohammadzadeh, A., Sabzalian, M. H., and Zhang, W., "An interval type-3 fuzzy system and a new online fractional-order learning algorithm: Theory and practice", IEEE Trans. Fuzzy Syst., vol. 28, no. 9, pp. 1940–1950, Sep. 2020.
10. Liu, Z., Mohammadzadeh, A., Turabieh, H., Mafarja, M., Band S. S. and Mosavi, A., "A New Online Learned Interval Type-3 Fuzzy Control System for Solar Energy Management Systems," in *IEEE Access*, vol. 9, pp. 10498-10508, 2021.
11. Singh, D., Verma, N. K., Ghosh, A. K., & Malagaudanavar, A. K. (2021). An Approach Towards the Design of Interval Type-3 TS Fuzzy System. *IEEE Transactions on Fuzzy Systems*.
12. Wang, J. H., Tavoosi, J., Mohammadzadeh, A., Mobayen, S., Asad, J. H., Assawinchaichote, W., & Skruch, P. (2021). Non-Singleton Type-3 Fuzzy Approach for Flowmeter Fault Detection: Experimental Study in a Gas Industry. *Sensors*, *21*(21), 7419.
13. Alattas, K. A., Mohammadzadeh, A., Mobayen, S., Aly, A. A., & Felemban, B. F. (2021). A New Data-Driven Control System for MEMSs Gyroscopes: Dynamics Estimation by Type-3 Fuzzy Systems. *Micromachines*, *12*(11), 1390.
14. Cao, Y., Raise, A., Mohammadzadeh, A., Rathinasamy, S., Band, S. S., & Mosavi, A. (2021). Deep learned recurrent type-3 fuzzy system: Application for renewable energy modeling/prediction. *Energy Reports*, *7*, 8115-8127.

15. Amador-Angulo, L.; Castillo, O.; Castro, J.; Melin, P. A New Approach for Interval Type-3 Fuzzy Control of Nonlinear Plants, *Int. J. Fuzzy Syst.* **2023**, *25*, 1624–1642.
16. Castillo, O., Valdez, F., Melin P. and Ding, W., A Survey on Type-3 Fuzzy Logic Systems and Their Control Applications, in *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 8, pp. 1744–1756, August 2024, https://doi.org/10.1109/JAS.2024.124530.
17. Ochoa, P.; Peraza, C.; Melin, P.; Castillo, O.; Park, S.; Geem, Z.W. Enhancing Control Systems through Type-3 Fuzzy Logic Optimization. *Mathematics* **2024**, *12*, 1792. https://doi.org/10.3390/math12121792.
18. Castillo, O., Melin, P. (2023). Type-3 Fuzzy Theory. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_2.
19. Castillo, O., Melin, P. (2024). Prediction with a Hybrid Interval Type-3 Fuzzy-Fractal Approach. In: Type-3 Fuzzy Logic in Time Series Prediction. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-59714-5_4.
20. Castillo, O., Melin, P. (2024). Type-3 Fuzzy Aggregation of Neural Networks. In: Type-3 Fuzzy Logic in Time Series Prediction. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-59714-5_5.
21. Castillo, O., Melin, P. (2023). Type-3 Fuzzy Control. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_1.
22. Castillo, O., Melin, P. (2023). Review of Type-3 Fuzzy Control. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_3.
23. Castillo, O., Melin, P. (2024). Towards Designing Interval Type-3 Fuzzy PID Controllers. In: Castillo, O., Melin, P. (eds) New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics. Studies in Computational Intelligence, vol 1149. Springer, Cham. https://doi.org/10.1007/978-3-031-55684-5_4.
24. Castillo, O., Melin, P. (2024). Towards Interval Type-3 Fuzzy Parameter Adaptation in Metaheuristics. In: Melin, P., Castillo, O. (eds) New Directions on Hybrid Intelligent Systems Based on Neural Networks, Fuzzy Logic, and Optimization Algorithms. Studies in Computational Intelligence, vol 1146. Springer, Cham. https://doi.org/10.1007/978-3-031-53713-4_1.
25. Ballinas, E.; Montiel, O.; Martínez-Vargas, A.; Rodríguez-Cortés, G. Hybrid Quantum Genetic Algorithm with Fuzzy Adaptive Rotation Angle for Efficient Placement of Unmanned Aerial Vehicles in Natural Disaster Areas. *Axioms* **2024**, *13*, 48. https://doi.org/10.3390/axioms13010048
26. Montiel, O., A Review of Quantum-Inspired Metaheuristics: Going from Classical Computers to Real Quantum Computers, in *IEEE Access*, vol. 8, pp. 814-838, 2020, https://doi.org/10.1109/ACCESS.2019.2962155.
27. L Astudillo, O Castillo, P Melin, A Alanis, J Soria, LT Aguilar, Intelligent Control of an Autonomous Mobile Robot using Type-2 Fuzzy Logic, Engineering Letters 13 (3), 2006.
28. R Sepúlveda, O Montiel, O Castillo, P Melin, Embedding a high speed interval type-2 fuzzy controller for a real plant into an FPGA, *Applied Soft Computing* 12 (3), 988-998, 2012.

Review on Quantum Optimization of Neural Networks and Fuzzy Systems

Patricia Melin

Abstract In this paper we are presenting an overview of the existing works that have been published in the quantum optimization of neural networks and fuzzy systems, considering both theoretical contributions and applications. The analysis was done considering all kinds of publications: journal papers, chapters in books and conference papers. The database was obtained from Scopus system of Elsevier, which contains most of the indexed publications in the world in all areas. Recently, type-2 and type-3 fuzzy systems have been used to aggregate neural networks in forming ensemble models, but their optimization is challenging, so we expect that quantum optimization could be helpful in this area. Based on the obtained data, relevant conclusions were obtained about the development of the quantum optimization of neural networks and fuzzy systems, as well as future trends for this evolving field.

Keywords Quantum optimization · Neural networks · Fuzzy systems

1 Introduction

Fuzzy logic since its inception [1] has been successfully applied in a plethora of areas [2, 3]. The main idea of fuzzy logic was to model linguistic uncertainty in real human decision-making, and the original fuzzy logic is now called type-1 as it was the initial version of this idea. More recently, type-2 was proposed and developed to offer a better model of uncertainty [4] and consequently more complex problems were solved [5, 6]. Still, more complex problems can be tackled with a most recent version of fuzzy theory, called type-3 fuzzy logic [7–9], which offers an even better model of uncertainty a now some recent successful applications have been achieved, like in [10–17].

These previous works consider problems from different perspectives, but in general the decision-making process is viewed as a problem under uncertainty. For

P. Melin (✉)
Tijuana Institute of Technology, TecNM, Tijuana, BC, Mexico
e-mail: pmelin@tectijuana.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_13

this reason, we decided to consider analyzing the existing works utilizing type-3 fuzzy logic, as this approach offers in theory a more powerful way to represent uncertainty. Accordingly, the contribution is the study and analysis of type-3 fuzzy literature. We can mention that there is currently some recent success in handling prediction with type-3, like in [18–20]. Also, there are some good results in control with type-3, like in [21–23]. In addition, the application of type-3 fuzzy logic in enhancing metaheuristics, like in [24]. The aim of type-3 in all cases is to capture in a better way the involved uncertainty in decision-making. The main problem in applying fuzzy systems is the finding their optimal structure and parameters. This is where quantum optimization techniques, as in [25, 26], can help in obtaining better fuzzy systems for the applications. The problem is even more complex for higher types of fuzzy systems, like type-2 and type-3, and this is where we believe quantum optimization could help in enhancing the results.

Neural Networks are models that simulate the human brain behavior and are formed the topology of the network and a learning algorithm [27]. These models are built with real data and the goal is to be able to solve problems, such as in prediction, recognition and control. There are also hybrid neuro-fuzzy systems that combine the advantages of neural networks (NNs) and fuzzy systems (FSs). Recently, type-2 and type-3 fuzzy systems have been used to aggregate neural networks in forming ensemble models, but their optimization is challenging, so we expect that quantum optimization could be helpful in this area.

The rest of the paper is as follows. In Sect. 2 we offer some basic concepts of fuzzy systems. Section 3 describes a review of the existing papers on quantum optimization of NNs and FSs. Section 4 shows the analysis of results, and Sect. 5 offers the conclusions.

2 Fuzzy Systems

There exists different types of Fuzzy Sets, in this case, for example: Type-1 [1–3], Type-2 [28, 29] and Type-3 [7]. All of them with different capabilities for handling uncertainty and with different computational complexities. The definitions of Type-1, Type-2, and Type-3 fuzzy sets are expressed in (1), (2) and (3), respectively.

$$A^{(1)} = \{(x, \mu_A(x)) | \forall x \in [0, 1]\} \tag{1}$$

$$A^{(2)} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u)\right) \middle| \forall u \in J_x \subseteq [0, 1] \right\} \tag{2}$$

$$A^{(3)} = \left\{ \left((x, u), \mu_{A^{(3)}}(x, u, v)\right) \middle| x \in X,\ u \in U \subseteq [0, 1],\ v \in V \subseteq [0, 1] \right\} \tag{3}$$

As can be noted, Type-2 Fuzzy Sets (T2 FSs) and Type-3 Fuzzy Sets (T3 FSs) represent implementation challenges because these systems require a greater computational cost when compared with respect to Type-1 Fuzzy Sets (T1 FSs). Since a T2

FS can be understood as a collection of embedded Type-1 membership functions (T1 MFs) over the secondary axis u and Type-3 Fuzzy Sets are a collection of embedded Type-2 membership functions (T2 MFs) over the third axis v, it is clear that a higher computer overhead is required to compute them. Equations (1) to (3) illustrate the general models of the presented type of Fuzzy Sets. However, there exist simplified approaches of these models to handle different levels of uncertainty, such as Interval Type-2 Fuzzy Sets (IT2 FSs as a simplification of T2 FSs) and Interval Type-3 Fuzzy Sets (IT3 FSs as a simplification of T3 FSs).

3 Review on Quantum Optimization of Neural Networks and Fuzzy Systems

In this section we describe the summary of the search results in Scopus when we use the query "quantum optimization of neural networks and fuzzy systems", which produces 119 documents. Of this total, there are 33 journal papers, 63 review papers, 14 conference papers, 3 chapters in books, and one book. In Fig. 1 we illustrate the number of publications per year in this area. Table 1 shows the same information for the most recent years. Figure 2 illustrates the most prominent authors and Table 2 lists their number of publications. Figure 3 shows the documents produced by the institutions, where Liaoning University of Technology and Jiangnan University of China are the top institutions in this area.

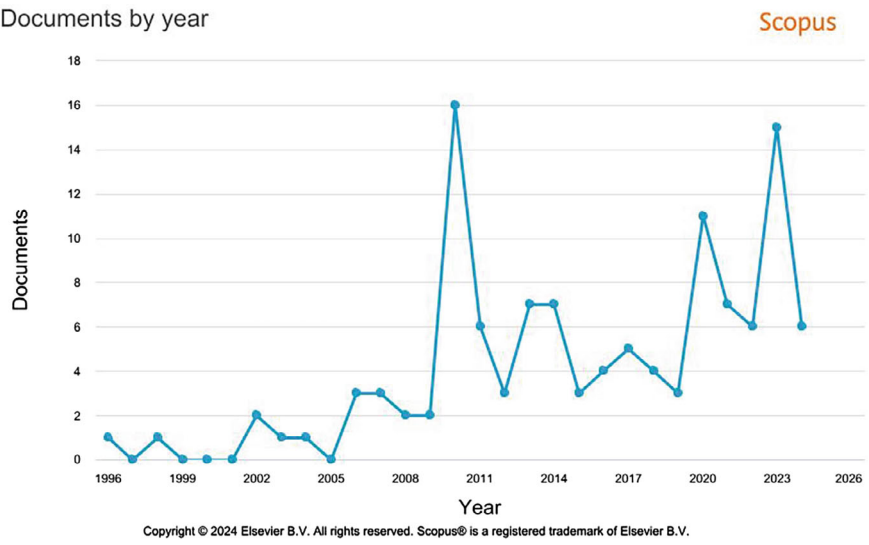


Fig. 1 Publications per year on quantum optimization of NNs and FSs

Table 1 Publications per year in quantum optimization of NNs and FSs

Year	Documents
2024	6
2023	15
2022	6
2021	7
2020	11
2019	3
2018	4
2017	5
2016	4

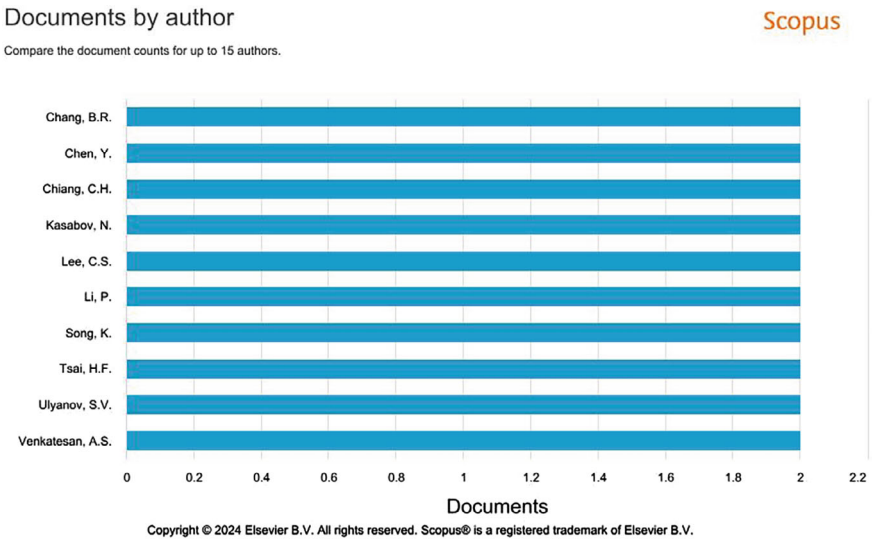


Fig. 2 Most prominent authors on quantum optimization of NNs and FSs

Table 2 Authors in quantum optimization of NNs and FSs

Author	Documents
Chang., B.R.	2
Chen, Y.	2
Chiang, C.H.	2
Kasabov, N.	2
Lee, C.S.	2
Li, P.	2
Song, K.	2
Tsai, H.F.	2
Ulyanov, S.V.	2

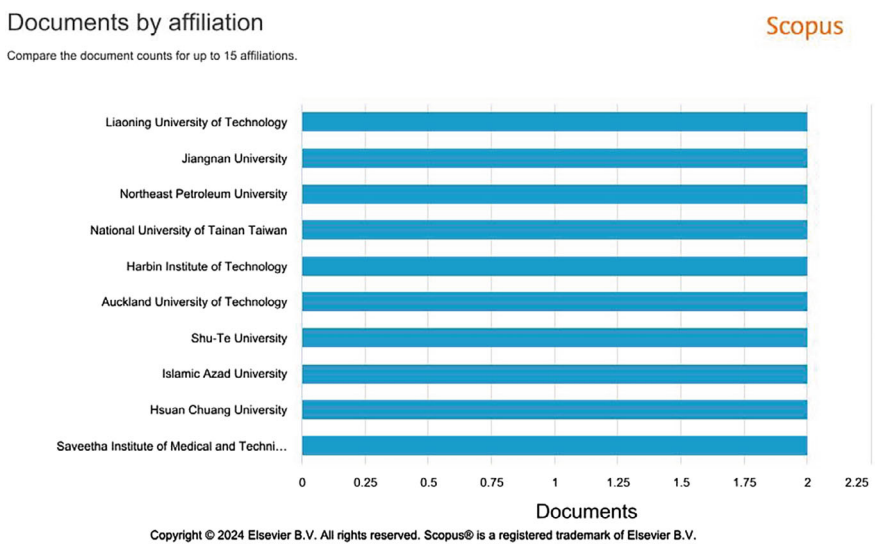


Fig. 3 Top institutions on quantum optimization of NNs and FSs

Figure 4 illustrates the most productive countries in the area of quantum optimization of NNs and FSs. Table 3 list the respective countries with the most publications in this area.

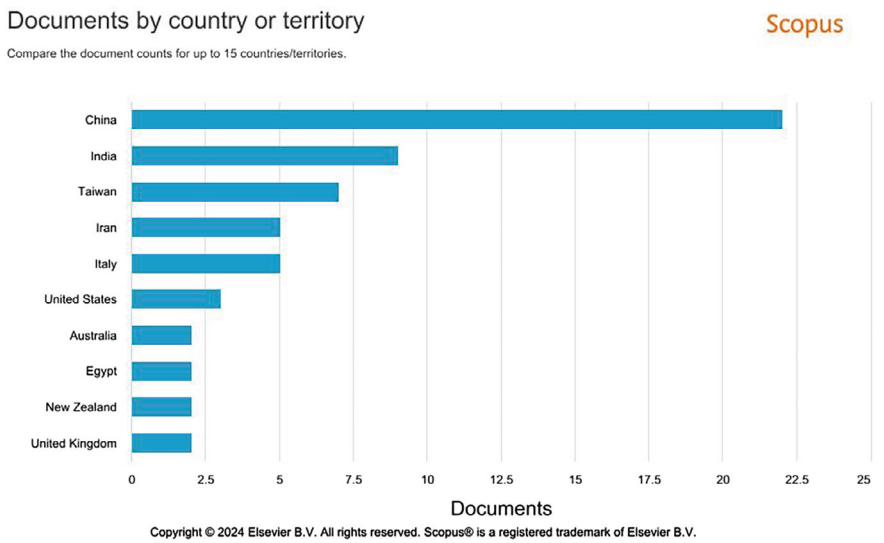


Fig. 4 Most productive countries in the area

Table 3 Most productive countries in quantum optimization of neural networks and fuzzy systems

Country	Documents
China	22
India	9
Taiwan	7
Iran	5
Italy	5
United States	3
Australia	2
Egypt	2
New Zealand	8

Figure 5 illustrates the distribution of published documents according to their type with conference reviews being the most prominent (52.9% of the total) and followed by articles in journals (27.7% of the total. Table 4 lists the corresponding information about the publications.

Figure 6 exhibits the distribution of papers by subject area, in which the most prominent area is computer science, followed by engineering and mathematics. Table 5 summarizes the publications according to their respective area.

Finally, Fig. 7 illustrates the top funding sponsors for quantum optimization of NNs and FSs. The top sponsor is National Natural Science Foundation of China.

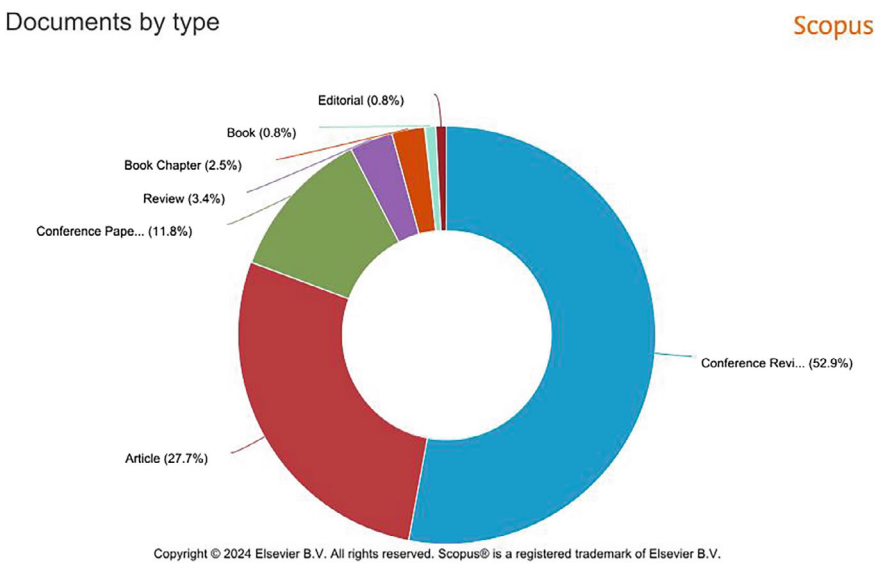


Fig. 5 Distribution of published documents according to their type

Table 4 Distribution of publications in quantum optimization of NNs and FSs

Author	Documents
Conference review	63
Article	33
Conference paper	14
Review	4
Book chapter	3
Editorial	1
Book	1

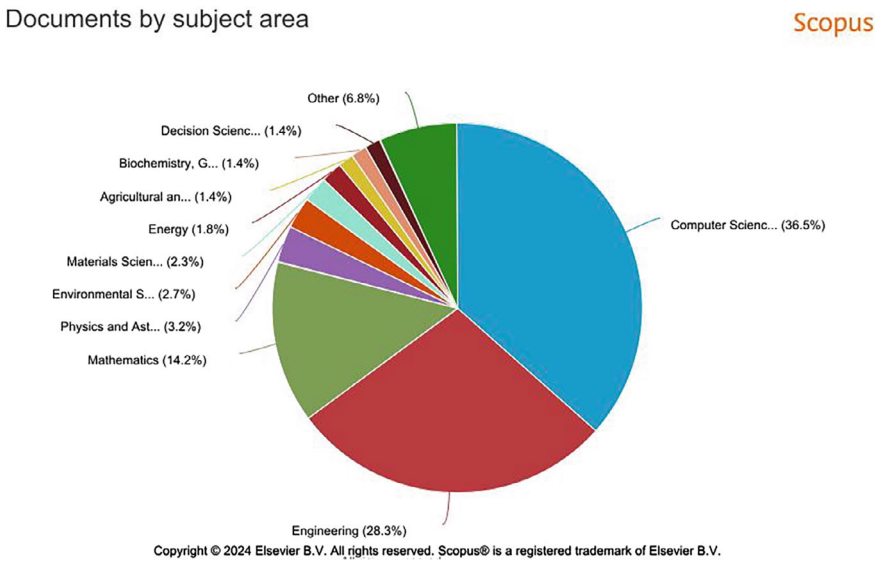


Fig. 6 Distribution of published documents according to their subject area

Table 5 Distribution of publications in quantum optimization of NNs and FSs according to the area

Area	Documents
Computer science	80
Engineering	62
Mathematics	31
Physics and astronomy	7
Environmental science	6
Material sciences	5
Energy	4
Agricultural and biological sciences	3
Biochemistry, Genetics and Molecular Biology	3

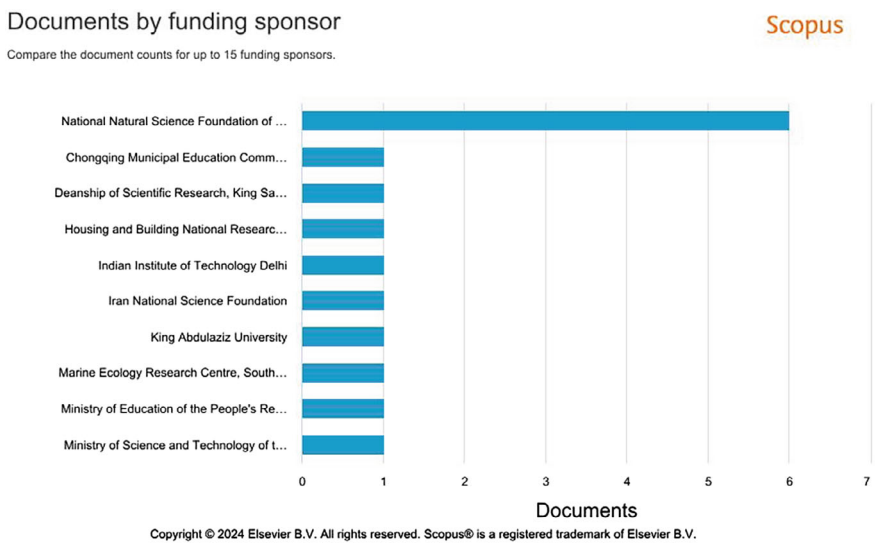

Documents by funding sponsor
Compare the document counts for up to 15 funding sponsors.
Scopus
National Natural Science Foundation of ...
Chongqing Municipal Education Comm...
Deanship of Scientific Research, King Sa...
Housing and Building National Researc...
Indian Institute of Technology Delhi
Iran National Science Foundation
King Abdulaziz University
Marine Ecology Research Centre, South...
Ministry of Education of the People's Re...
Ministry of Science and Technology of t...
0
1
2
3
4
5
6
7
Documents
Copyright © 2024 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Fig. 7 Distribution of published documents according to their funding sponsor

4 Analysis and Future Trends for Quantum Optimization of Neural Networks and Fuzzy Systems

In this section we describe the analysis of the data obtained from Scopus and also, we envision possible future trends for the area of quantum optimization of NNs and FSs. Based on the publications that have been made, we can say that most of the papers on quantum optimization of NNs and FSs have been applied in the computer science and engineering areas. We did notice that at the moment only fuzzy systems of type-1 and type-2 have been optimized with quantum algorithms, but we consider that type-3 will be more prominent in the future, as type-3 fuzzy systems have more parameters and are more complex. In addition, another topic that has not been explored is the use of fuzzy systems for achieving dynamic parameter adaptation in quantum algorithm, which could help in achieving even better results in many complex problems. Finally, there are also theoretical work on type-n fuzzy logic, which could also allow testing with quantum optimization on type-n fuzzy systems in the near future.

5 Conclusions

In this paper we have outlined a review of the publications on quantum optimization of neural networks and fuzzy systems in recent years, which include papers in journals, conference proceeding and in edited books. The statistics of publications, authors and institutions were presented, as well as the analysis of the data and future trends

for the area of quantum optimization of neural networks and fuzzy systems. Based on the results and analysis we believe that this area is becoming a very important topic, and we expect that in the future more theoretical advances will come, as well as new applications will be developed.

References

1. Zadeh L.A. (1965) Fuzzy sets. Inf Control 8:338–353.
2. Zadeh, L. A., Knowledge representation in Fuzzy Logic, IEEE Transactions on knowledge data engineering, Vol. 1, p.89, 1989.
3. Zadeh, L. A., The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* **1975**, *8*, 199-249.
4. Mendel JM, Bob John RI (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127.
5. Ontiveros, E.; Melin, P.; Castillo, O. Comparative study of interval Type-2 and general Type-2 fuzzy systems in medical diagnosis. *Inf. Sci.* **2020**, *525*, 37–53.
6. Ontiveros-Robles, E.; Castillo, O.; Melin, P. Towards asymmetric uncertainty modeling in designing General Type-2 Fuzzy classifiers for medical diagnosis. *Expert Syst. Appl.* **2021**, *183*, 115370.
7. Castillo, O., Castro, J.R., Melin, P., Interval Type-3 Fuzzy Systems: Theory and Design, Springer, Cham, Switzerland, 2022.
8. Qasem S.N., Ahmadian A., Mohammadzadeh A., Rathinasamy S., Pahlevanzadeh B. A type-3 logic fuzzy system: Optimized by a correntropy based Kalman filter with adaptive fuzzy kernel size, Inform. Sci., 572 (2021), pp. 424-443
9. Mohammadzadeh, A., Sabzalian, M. H., and Zhang, W., "An interval type-3 fuzzy system and a new online fractional-order learning algorithm: Theory and practice", IEEE Trans. Fuzzy Syst., vol. 28, no. 9, pp. 1940–1950, Sep. 2020.
10. Liu, Z., Mohammadzadeh, A., Turabieh, H., Mafarja, M., Band S. S. and Mosavi, A., "A New Online Learned Interval Type-3 Fuzzy Control System for Solar Energy Management Systems," in *IEEE Access*, vol. 9, pp. 10498-10508, 2021.
11. Singh, D., Verma, N. K., Ghosh, A. K., & Malagaudanavar, A. K. (2021). An Approach Towards the Design of Interval Type-3 TS Fuzzy System. *IEEE Transactions on Fuzzy Systems*.
12. Wang, J. H., Tavoosi, J., Mohammadzadeh, A., Mobayen, S., Asad, J. H., Assawinchaichote, W., & Skruch, P. (2021). Non-Singleton Type-3 Fuzzy Approach for Flowmeter Fault Detection: Experimental Study in a Gas Industry. *Sensors*, *21*(21), 7419.
13. Alattas, K. A., Mohammadzadeh, A., Mobayen, S., Aly, A. A., & Felemban, B. F. (2021). A New Data-Driven Control System for MEMSs Gyroscopes: Dynamics Estimation by Type-3 Fuzzy Systems. *Micromachines*, *12*(11), 1390.
14. Cao, Y., Raise, A., Mohammadzadeh, A., Rathinasamy, S., Band, S. S., & Mosavi, A. (2021). Deep learned recurrent type-3 fuzzy system: Application for renewable energy modeling/prediction. *Energy Reports*, *7*, 8115-8127.
15. Amador-Angulo, L.; Castillo, O.; Castro, J.; Melin, P. A New Approach for Interval Type-3 Fuzzy Control of Nonlinear Plants, *Int. J. Fuzzy Syst.* **2023**, *25*, 1624–1642.
16. Castillo, O., Valdez, F., Melin P. and Ding, W., A Survey on Type-3 Fuzzy Logic Systems and Their Control Applications, in *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 8, pp. 1744-1756, August 2024, https://doi.org/10.1109/JAS.2024.124530.
17. Ochoa, P.; Peraza, C.; Melin, P.; Castillo, O.; Park, S.; Geem, Z.W. Enhancing Control Systems through Type-3 Fuzzy Logic Optimization. *Mathematics* **2024**, *12*, 1792. https://doi.org/10.3390/math12121792.

18. Castillo, O., Melin, P. (2023). Type-3 Fuzzy Theory. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_2.
19. Castillo, O., Melin, P. (2024). Prediction with a Hybrid Interval Type-3 Fuzzy-Fractal Approach. In: Type-3 Fuzzy Logic in Time Series Prediction. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-59714-5_4.
20. Castillo, O., Melin, P. (2024). Type-3 Fuzzy Aggregation of Neural Networks. In: Type-3 Fuzzy Logic in Time Series Prediction. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-59714-5_5.
21. Castillo, O., Melin, P. (2023). Type-3 Fuzzy Control. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_1.
22. Castillo, O., Melin, P. (2023). Review of Type-3 Fuzzy Control. In: Type-3 Fuzzy Logic in Intelligent Control. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-46088-3_3.
23. Castillo, O., Melin, P. (2024). Towards Designing Interval Type-3 Fuzzy PID Controllers. In: Castillo, O., Melin, P. (eds) New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics. Studies in Computational Intelligence, vol 1149. Springer, Cham. https://doi.org/10.1007/978-3-031-55684-5_4.
24. Castillo, O., Melin, P. (2024). Towards Interval Type-3 Fuzzy Parameter Adaptation in Metaheuristics. In: Melin, P., Castillo, O. (eds) New Directions on Hybrid Intelligent Systems Based on Neural Networks, Fuzzy Logic, and Optimization Algorithms. Studies in Computational Intelligence, vol 1146. Springer, Cham. https://doi.org/10.1007/978-3-031-53713-4_1.
25. Ballinas, E.; Montiel, O.; Martínez-Vargas, A.; Rodríguez-Cortés, G. Hybrid Quantum Genetic Algorithm with Fuzzy Adaptive Rotation Angle for Efficient Placement of Unmanned Aerial Vehicles in Natural Disaster Areas. *Axioms* **2024**, *13*, 48. https://doi.org/10.3390/axioms13010048
26. Montiel, O., A Review of Quantum-Inspired Metaheuristics: Going From Classical Computers to Real Quantum Computers, in *IEEE Access*, vol. 8, pp. 814-838, 2020, https://doi.org/10.1109/ACCESS.2019.2962155.
27. Melin, P., Castillo, O. Supervised Learning Neural Networks. In: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing. Studies in Fuzziness and Soft Computing, vol 172. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-32378-5_4.
28. Astudillo, L., Castillo, O., Melin, P., Alanis, A., Soria, J., Aguilar, L.T., Intelligent Control of an Autonomous Mobile Robot using Type-2 Fuzzy Logic, Engineering Letters 13 (3), 2006.
29. Sepúlveda, R., Montiel, O., Castillo, O., Melin, P., Embedding a high speed interval type-2 fuzzy controller for a real plant into an FPGA, *Applied Soft Computing* 12 (3), 988-998, 2012.

Finance in the Quantum Computing Era: A Survey from 2019 to 2024

Luis Enrique Andrade-Gorjoux, César Castrejón-Peralta, Jorge Saavedra-Benavides, Jesús Yaljá Montiel-Pérez, and Eduardo Bustos-Farias

Abstract Finance in quantum computing is an emerging area of research that explores how this new technology can transform the financial industry. The promise of this approach lies in its ability to solve problems where classical computers have been unable to offer efficient solutions due to the nature of financial problems, which are often intractable by classical algorithms. The current challenges are in various areas of application such as optimization when handling large amounts of information and its constraints, implementing alternatives to valuation models, proposing new approaches in areas of quantum machine learning, quantum prediction, among others. To overcome current challenges and take advantage of new emerging technologies, it is crucial to foster collaboration between the financial sector, academia and the technology industry, reviewing and addressing the potential of these technologies, such as quantum computing, in everyday areas such as the field of finance.

1 Introduction

The area of finance is responsible for managing the economic resources of an organization or individual, with the objective of maximizing their value and minimizing the

L. E. Andrade-Gorjoux · C. Castrejón-Peralta (✉) · J. Saavedra-Benavides · J. Y. Montiel-Pérez
Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México, Mexico
e-mail: ccastrejonp2021@cic.ipn.mx

L. E. Andrade-Gorjoux
e-mail: luis.gorjoux@gmail.com; landradeg2022@cic.ipn.mx

J. Saavedra-Benavides
e-mail: jsaavedrab2024@cic.ipn.mx

J. Y. Montiel-Pérez
e-mail: jyalja@cic.ipn.mx

E. Bustos-Farias
Centro de Investigaciones Económicas Administrativas y Sociales, Instituto Politécnico Nacional, Ciudad de México, Mexico
e-mail: ebustosf@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_14

associated risks [80]. In this sense, this field encompasses a wide range of activities, from investment management and risk analysis to asset valuation and financial planning [2]. Today, this area has become specialized, requiring the use of advanced computational tools and techniques to address the challenges and opportunities presented by today's economic environment [3].

The convergence point between finance and quantum computing represents a field with the ability to transform contemporary financial technology [74]. As we enter the third decade of the 21st century, the convergence of these two domains is generating a new paradigm known as *quantum finance*, which promises to revolutionize the way challenges are addressed and strategic decisions are made in the financial sector [4].

Quantum computing, based on the principles of quantum mechanics [10–12], offers a radically different approach to processing and working information with classical computational systems, as discussed by [14, 21]. By taking advantage of quantum phenomena such as superposition and entanglement, quantum computers have the potential to solve certain types of problems more efficiently than their classical counterparts [2], following the implementation of various computational algorithms with a quantum approach [23–25] or those seeking a hybrid version, taking advantage of the best of both computational approaches [29–31].

This sector, characterized by its dynamic nature, is uniquely positioned to benefit from advances in quantum computing, as [15, 16] comment, by processing massive volumes of data, simulating market scenarios and optimizing investment strategies capturing the attention of financial institutions, regulators and academics alike [5].

Technical hurdles, mentioned by [8, 26–28], such as the need for more stable and scalable quantum systems, persist. In addition, the shortage of talent with expertise in both this area and quantum physics presents a significant challenge to the industry.

The impact of quantum finance extends beyond incremental improvements in operational efficiency with the technology expected to have a significant impact on financial markets, disrupting existing business models and creating new opportunities for innovation [6]. In addition, the intersection of quantum computing with other emerging technologies, such as artificial intelligence and blockchain, may create new financial products and services, as well as novel approaches to market regulation and oversight [9].

In the following sections, we examine the fundamentals of quantum computing and its relevance to finance, current and potential applications in various areas of the financial sector, as well as the technical and practical challenges facing the field, finally presenting a brief discussion of the implications of the use of this emerging technology for the future of global finance.

1.1 Search Methodology

Given that quantum computing and studies on quantum finance are novel and rapidly evolving fields, this analysis focuses on recent publications, considering only those from 2019 to 2024.

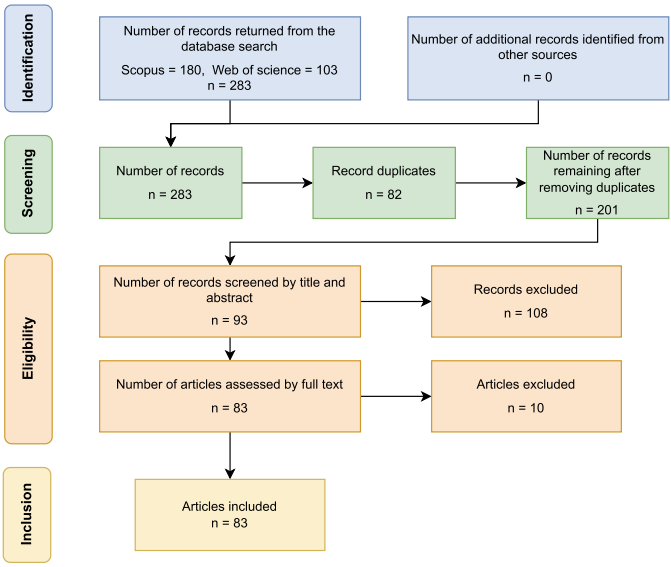


Fig. 1 PRISMA model of literature review process

A summary of the process for identifying and selecting articles is presented in Fig. 1, that was created following the PRISMA methodology [7].

Based on this analysis, the following research questions were proposed:

- What are the fundamentals of quantum computing and how are they applied in the field of finance?
- What financial applications can be addressed using quantum computing?
- What quantum methods and algorithms have been proposed to solve financial problems?
- What are the challenges and limitations in the use of quantum algorithms in finance?

Besides the period of time, there were only considered research and conference papers published in english and spanish listed in Scopus and Web of Science portals, considering specific keywords, such as *quantum finance, quantum computing, quantum optimization* and *quantum machine learning*.

2 Emerging Applications

The financial sector is characterized by its inherent complexity, dynamism and the need to process large volumes of data in real time. This unique nature presents a number of computational challenges that, in many cases, test the limits of classical computing [32].

Quantum computing, with its ability to process information in a fundamentally different way than classical computing, is emerging as a promising technology to address some of the most pressing challenges in the financial sector [47, 48].

In this section, we examine the nature of some important financial problems and the limitations faced by traditional computational approaches in addressing them, as well as the emerging applications of quantum computing in finance, highlighting how this innovative technology could transform various aspects of these situations.

2.1 Portfolio Optimization

Portfolio optimization is a fundamental problem in finance that seeks to maximize expected return while minimizing risk of lost [33]. Modern portfolio theory, developed by Harry Markowitz [34], laid the foundation for the field; however, as financial markets have become more complex and globalized, portfolio optimization has evolved into a computationally intensive problem [35].

The nature of the portfolio optimization problem is inherently combinatorial in that, as the number of assets considered increases, the solution space grows exponentially [36]. Moreover, the incorporation of constraints, such as investment limits, transaction costs and liquidity considerations, increases the complexity of the problem. Classical algorithms, such as quadratic programming, face significant limitations when dealing with large-scale portfolios with multiple objectives and constraints [38].

Classical computation faces a bottleneck in attempting to exhaustively explore the vast solution space in a reasonable amount of time. This has led to the adoption of heuristics and approaches that, while practical, may not achieve global optimal optimization [37].

In [50] it is mentioned that the Quantum Support Vector Machines (QSVM) algorithm has shown significant superiority over algorithms such as Quantum Principal Component Analysis (QPCA), Quantum Boltzmann Machines (QBM) and Quantum K-means (QKMeans) algorithm in portfolio optimization.

Another approach proposed in [49] considers an algorithm based on quantum walking, presents a theoretical approach that is implemented experimentally using optical lattices and single photons. After analysis, he confirms that the algorithm amplifies the probability of finding attractive financial solutions by increasing the number of iterations.

An optimization algorithm called Quantum Beetle Antenna Search (QBAS) for investment portfolio selection is presented in [1]. QBAS is a variant of the classical Beetle Antenna Search (BAS) algorithm adapted to quantum computing. The results show that QBAS outperforms other metaheuristic algorithms such as the particle swarm optimization (PSO) algorithm and the genetic algorithm (GA).

Similarly, [51] proposes an intelligent investment system combining Deep Reinforcement Learning (DRL) and Quantum Finance Theory. The system consists of two agents: the first is a deterministic deep policy gradient (DDPG)-based trading

agent that defines the investment ratio and the second is a policy gradient (PG)-based risk control agent that uses daily quantum price levels (QPL) to make discrete decisions. The collaboration between the two agents allows for more reasonable and stable investment settings. Back-testing results with FOREX products demonstrate the flexibility and robustness of the system, which generates considerable profits. In addition, [52] reaffirms the use of DDPG following the generation of the Deep Reinforcement Learning (DRL) model to dynamically optimize investment portfolios, which improves learning speed and reduces data requirements compared to previous models. It also presents a novel risk control based on Quantum Price Levels (QPLs) from Quantum Finance Theory that shows that the model obtains higher returns with lower risk and data requirements compared to previous models.

In [40], its proposed a quantum algorithm to calculate the risk contribution of subgroups within a credit portfolio, complementing existing methods for measuring overall risk. Unlike the classical method, the quantum method offers an advantage for high-precision calculations, being even less complex in realistic scenarios; however, for analyzing highly divided subgroups, its benefit decreases compared to the calculation of the total portfolio risk.

From these examples, it is possible to observe some success stories in how quantum computing is revolutionizing portfolio optimization, allowing the efficient exploration of solutions in a high-dimensional space while considering complex constraints. These advances have the potential to significantly improve investment management and financial decision-making, opening new opportunities for innovation and value creation in the financial sector.

2.2 Derivatives Pricing

Accurate valuation of derivative financial instruments is crucial to the stability of the financial system. These problems often involve solving complex partial differential equations and simulating multiple market scenarios [39].

The Monte Carlo method, widely used in derivatives pricing, requires a high number of simulations to achieve accurate results, especially for complex, volatile and atypical financial products used in the investment area [42]. The sequential nature of many classical simulation algorithms limits the ability to take full advantage of parallelism in traditional computing architectures.

As mentioned by [54], in their study on quantum computing applied to derivatives valuation and risk estimator calculations such as VaR (Value at Risk), while quantum algorithm implementations show promising advances, further research is still required for practical applications in the financial industry.

In [55], a method based on quantum principal component analysis (QPCA) is proposed to reduce the complexity of the Heath-Jarrow-Morton model used in derivatives pricing. This method was experimentally implemented on a five-qubit IBMQX2 quantum computer, yielding promising results for the development of more general quantum algorithms applicable to derivative valuation.

On the other hand, quantum algorithms have the potential to significantly accelerate Monte Carlo simulations, a widely used method in derivative pricing, following the development of the Variational Quantum Monte Carlo (VMC) model and neural networks to address the complexity of time-dependent partial differential equations (PDEs) [41]. This technique is demonstrated in the multi-asset Black-Scholes equation for valuing complex financial options. Although these equations often lack analytical solutions and require complex numerical methods, as noted by [56], a framework is proposed that consists of two stages: data preparation, which exponentially compresses the information, and information extraction, which decodes it with considerable acceleration. This reduction in complexity allows for the analysis of complex stochastic processes, including extreme events and historical information, proving versatile with applications in finance and other fields, making it a powerful tool for precise analysis of random phenomena.

These advances highlight how quantum computing is transforming derivatives pricing by enabling the efficient simulation of multiple market scenarios and the accurate valuation of complex financial instruments. These developments have the potential to enhance transparency and efficiency in derivative markets, as well as to drive innovation in the creation of new financial products.

2.3 Financial Risk Management

Real-time risk management, which involves continually re-evaluating positions and updating risk models, presents significant computational challenges [46]. Classic systems often struggle to maintain processing speed with the volume of market data, which can result in suboptimal or outdated risk assessments [45].

Within financial risks, it is possible to mention the accurate modeling of credit risk which is crucial to the stability of the financial system and requires consideration of a wide range of factors, from individual financial data to macroeconomic trends. The nature of this problem is inherently multidimensional and nonlinear and its a critical component in decision-making within the financial sector, which could use benefits from quantum computing in several ways [44].

In [45] is discussed the feasibility of using quantum computers to calculate market VaR (risk-to-value) and PFE (potential future exposure) of counterparty credit risk. While conceptual modeling and small quantum circuits are possible, the current capacity of quantum devices (around 100 qubits) is insufficient to handle thousands of real-life risk factors. Quantum noise also affects the accuracy of estimates. In addition, there are difficulties in loading realistic probability distributions into quantum registers and representing complex payoff functions. For quantum computing to be a viable option in financial risk measurement, significant advances in hardware, error control and noise mitigation are needed.

On the other hand, [46] proposes a new method for predicting credit problems based on quantum machine learning (QML) using Quantum Processing Units (QPU). This method combines classical algorithms with quantum computing to create faster,

more interpretable, and competitive models compared to existing ones. The model was tested with real data and compared to a classical reference model ("Random Forest"), showing similar precision with greater interpretability and requiring less training time. Although it cannot yet be executed with current quantum hardware, it outperforms the classical reference model, suggesting that such solutions could surpass current classical methods used for financial fraud prevention.

Classical methods for credit risk analysis (CRA) limit the process to one risk factor per asset, which reduces its ability to represent real situations. Therefore, [44] proposed a model that allows considering multiple systemic risk factors and provides a more accurate picture of the probability of default of each asset in a portfolio. In addition, it makes data input more flexible by allowing non-integer values (expected loss in case of default), bringing the model closer to real financial data. Its performance has been tested on classical simulators and on real quantum hardware (IBM Quantum Experience).

Another risk related to the field of finance is transaction security. Financial cybersecurity refers to the application of quantum computing in data protection and cyber threat detection in financial environments [68].

In the study by [57], an analysis of the impact of Artificial Intelligence (AI) on personalized banking is presented, an area of DARQ technologies (distributed accounting, AI, extended reality, quantum computing) with great potential in the financial sector. The results help understand how a global bank could leverage AI to launch a personalized mobile banking service, from identifying the target audience to designing the value proposition. The study also highlights risks such as data security and the wage gap that AI might create in the labor market. Despite this, it concludes that AI in mobile banking can be beneficial for both banks and customers.

Furthermore, [58] presents a model for authentication in cloud storage and data transfer based on secure key distribution and post-quantum cryptography, allowing advancements in cybersecurity and providing solutions to security problems in cloud systems, which handle many financial transactions. The model uses post-quantum algorithms to generate keys and protect data in transit, storage, and use, showing key generation and encryption times under 10 ms at quantum level 2, indicating high performance.

These advances can improve the resilience and stability of the financial system, as well as drive innovation in risk management and decision making in the financial sector.

2.4 Financial Fraud Detection

Fraud and anomaly detection in financial transactions is a critical problem that requires real-time analysis and classification of specific patterns in massive volumes of highly unbalanced data, i.e., composed mainly of legitimate transactions, with a very low percentage of anomalous patterns [18].

Classical approaches, which often rely on predefined rules or statistical models, may be insufficient to detect sophisticated and evolving fraud schemes. Machine learning techniques have improved detection capabilities, but training and implementing these models on large-scale financial datasets can be computationally intensive facing challenges in terms of speed and scalability when processing real-time financial data streams. In addition, the need to continuously update and adapt detection models to keep up with new fraud tactics increases the computational burden [17].

A QML algorithm for financial fraud detection using a qubit is proposed in [67]. The performance of a one-qubit quantum classifier is compared with its classical counterpart on a real data set of banking transactions (fraud or legitimate). Although the quantum classifier achieves similar accuracy to the classical model, the results do not prove an advantage of using quantum computation; however, the work shows the ability of quantum algorithms to solve problems in this area, allowing consideration of future architectures with more qubits that take advantage of quantum properties such as entanglement to improve fraud detection.

As discussed by [64] where he makes use of quantum learning based on quantum support vector machines (QSVM) for home insurance fraud detection. The QSVM model is compared with the classical SVM and feature engineering and parameter optimization techniques are explored to improve the accuracy of fraudulent policy detection, obtaining positive results for the quantum area. Likewise, [17] compares the performance of a classical machine learning model with a classifier based on quantum support vector machines (QSVM) for fraud detection in card payments, concluding that this method, together with the use of quantum entanglement, improves the overall performance.

Moreover, using cognitive computing in conjunction with quantum computing [65], it is possible to obtain a new method for detecting fraudulent transactions in cyber-physical banking systems (BCPS). The detection is based on the identification of suspicious entities and achieves an accuracy of 97.04% with an error rate of 0.03%, making this approach more reliable and efficient than classical methods.

Using quantum neural networks with graphs (QGNN), an algorithm for financial fraud detection was proposed in [66]. QGNNNs use quantum computation to process graph-structured data and outperform classical neural networks by using variational quantum circuits (VQCs) to improve performance. Experimental results show that QGNNs achieve 85% accuracy, outperforming classical neural networks with 77% accuracy, on a real financial fraud dataset. In addition, the research also analyzes the influence of the number of qubits and layers on the performance of QGNNs. It is observed that QGNNNs with 6 qubits and one layer achieve the best accuracy (94.5 %), due to the lower susceptibility to qubit noise and the ability of one layer to capture essential features of the data.

2.5 Time-Series Forecasting

Financial time series forecasting [75, 76], fundamental to investment decision making and risk management, presents unique challenges due to the non-linear and non-stationary nature of financial data. Traditional time series models, such as ARIMA and GARCH [43], present problems in capturing the dynamics of financial markets.

Machine learning and deep learning techniques have improved the ability to model complex relationships in financial data; however, these approaches often require large training datasets and significant computational resources [20]. In addition, the interpretability of deep neural models remains a challenge, which is crucial in a financial context where transparency and explainability are increasingly important. Likewise, there are limitations in processing high-dimensional financial datasets and capturing long-term dependencies in time series [19]. This can result in suboptimal predictions, especially in volatile market conditions or during extreme events.

Financial time series prediction, crucial for investment decision making and risk management, is another area where quantum computing promises significant advances [22].

Quantum algorithms have the potential to capture complex patterns and long-term dependencies in financial data that classical methods might miss; as explained by [59], after purchasing methods for forecasting global Gross Domestic Product (GDP) growth, proposing a novel model that combines quantum computing with deep learning, specifically with "Deep Neural Decision Trees", noting that the hybrid model of quantum computing and deep learning has a high potential to forecast GDP growth accurately at the global level, allowing policymakers to make better economic and monetary policy decisions.

In [60] is proposed a novel model for predicting the stock market that combines the classical hidden Markov model with quantum computing, which reduces the number of parameters required for prediction compared to the classical model. To verify this, experiments were carried out to simulate stock market trading for one week and a return of 6.7% on the initial capital was obtained. This supports the viability of the model, although improvements in data processing and verification methods are needed to be able to apply it to the market.

In addition, [61] proposes two hybrid classical-quantum architectures for predicting time series. Inspired by multilayer networks (MLP), these architectures combine variational quantum circuits with specific coding and optimization performed by a classical computer to evaluate prediction problems such as Mackey-Glass series, Lorenz attractors and Box-Jenkins series, and the USD-euro exchange rate in the area of finance, showing that the hybrid models (QNN and HQNNN) are competitive with MLP, CNN and LSTM models with a similar number of trainable parameters.

In [62], Quantum Neural Networks (QNN) are employed to predict inflation, and the results obtained show that QNN outperform existing models in terms of accuracy and can predict both the direction and magnitude of inflation. Moreover, QNNs perform better with CPI-core data (excluding food and energy), possibly due to the high price volatility of these commodities.

Furthermore, [19] makes use of this same type of neural networks for the prediction of various time series in areas such as finance and energy, showing that QNNs can capture long-term dependencies which makes them useful for price prediction and risk management.

In a more specific application case, [63] proposes a hybrid quantum learning model to predict carbon price. The L-QLSTM model combines quantum LSTM (QLSTM) with linear layers to enhance the learning capability of the quantum model. The linear layers are placed before and after the variational quantum circuits (VQCs) to extract features, reduce the use of qubits and enhance quantum advantages. In addition, parameters are shared in the linear layers and the VQC is optimized with higher quantum entanglement to improve learning. The results show that L-QLSTM outperforms classical LSTM and QLSTM models (Fig. 2) in terms of accuracy and efficiency, suggesting that the proposed hybrid model can help in price prediction.

Emerging applications of quantum computing in finance promise to fundamentally transform how complex financial problems are approached. From portfolio optimization to risk modeling, quantum computing offers the potential for faster, more accurate and sophisticated solutions. However, it is important to note that many of these applications are still in early stages of development and face significant challenges in terms of scalability and quantum coherence error. As quantum technology continues to mature, we are likely to see increasing adoption of these techniques in the financial industry, which could lead to more efficient markets, better risk management and new opportunities for financial innovation.

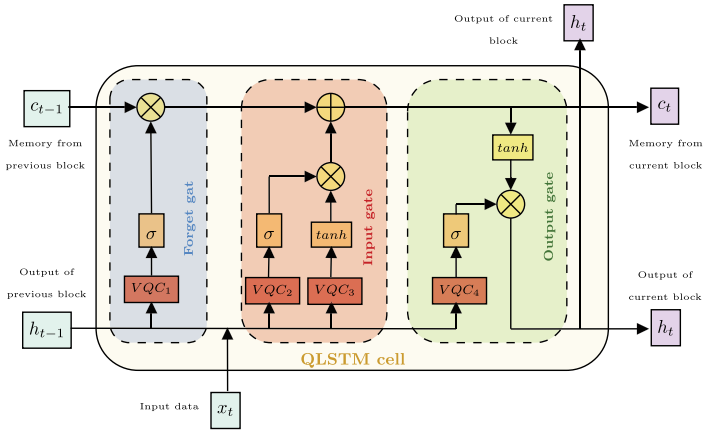


Fig. 2 Basic architecture of a quantum LSTM

3 Challenges of Quantum Computing in Finance

The implementation of quantum computing in the financial sector presents multiple challenges that must be overcome to fully exploit its potential. One of the main challenges is the adaptation of quantum algorithms to solve specific financial problems involving the development or modification of existing algorithms, requiring advanced knowledge in both financial theory and quantum mechanics to take advantage of properties such as superposition and entanglement (Fig. 3).

Another fundamental challenge lies in the integration of quantum technologies within current financial infrastructures, designed on years of advances in classical computing. Migrating to a structure that incorporates quantum elements requires not only technological advances, but also a significant restructuring of processes and systems, which will involve managing hybrid systems that combine classical and quantum components, generating considerable challenges in terms of data management, security and efficiency [51, 69, 70].

Data management in quantum environments also represents a complex challenge for the financial sector, as financial data, characterized by their volume and sensitivity, must be transformed into formats compatible with quantum systems without compromising their confidentiality and integrity due to the inherent uncertainty in quantum circuits, which produces probabilistic results [41, 53].

Despite the potential shown by quantum computer prototypes to solve small-scale financial problems, bringing these solutions to real problems faces significant limitations [71]. Portfolio optimization or market simulation, for example, require a large number of qubits and high-fidelity quantum operations, capabilities still limited in current quantum systems [55, 67].

Security and privacy are critical areas in the financial sector, and the inclusion of quantum technology raises new concerns, considering that one of the objectives stems from the need for post-quantum cryptography protocols to ensure that financial systems are protected against potential attacks of the same nature. In addition, there is a risk that quantum states can store more information than their classical counterparts, which increases vulnerability to cyberattacks [57, 68, 72].

To implement quantum computing in finance, it is essential to develop skills that combine quantum physics, computer science and financial theory. Education and collaboration between academic, technological and financial institutions are key to overcoming these challenges and creating educational programs that link quantum theory with practical applications in finance [73].

4 Financial Sector, Academy and Industry

The integration of quantum computing in the financial sector requires significant collaboration between financial institutions, academia and the technology industry. This synergy is essential to overcome the various technical and operational challenges

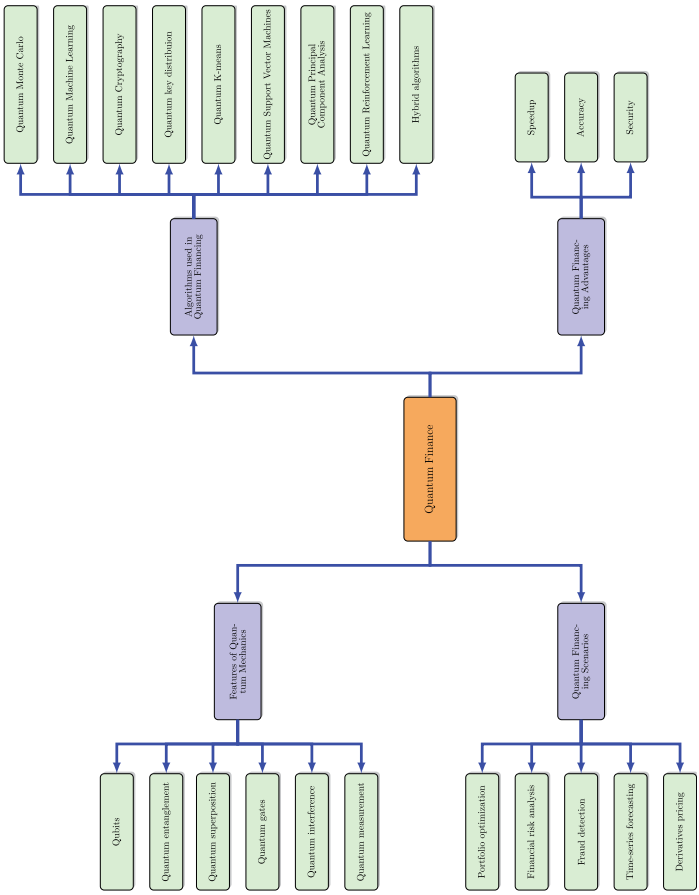


Fig. 3 The quantum financing system

that emerge in the adoption of quantum technologies in the financial arena, with each sector bringing its specific strengths to drive innovation [78].

Cyber risk management represents a critical concern in this context. In the work of [77], they emphasize the need to understand the frequency and severity of cyber attacks, highlighting the importance of implementing management programs that contemplate quantum cryptography. Their analysis underlines that collaboration between governments and financial institutions is essential to develop standardized frameworks for cyber incident management.

Collaborative research projects provide a significant boost to innovation in quantum finance, focusing on specific applications such as portfolio optimization and risk assessment, where financial institutions contribute sectoral knowledge and economic resources, while academia provides theoretical foundations and trains new experts. For its part, the technology industry contributes its expertise in quantum hardware and software development.

The shortage of specialized professionals has prompted the development of innovative educational programs that combine quantum physics, computer science and finance. Universities such as MIT, Stanford and Oxford are implementing interdisciplinary programs, while financial institutions are offering internships and specialized training.

Collaboration between finance, academia and industry also manifests itself in the production of joint publications and coordinated knowledge dissemination efforts. Research articles, white papers and technical reports that combine perspectives from the financial, academic and technological worlds are contributing significantly to the body of knowledge in quantum finance. These publications not only advance the state of the art of research, but also help educate a broader audience about the potential applications and implications of quantum computing in finance.

5 Conclusions

Quantum computing is emerging as a revolutionary element in the financial sector and is in a transitional phase between theory and practice. It is possible to observe throughout this chapter that, although quantum hardware has not yet completely outperformed traditional systems in complex financial problems, its development is progressing rapidly, while specific quantum algorithms are opening up new possibilities in portfolio optimization, pricing and risk management.

It is possible to consider that the impact of this technology goes beyond operational improvements, as it could fundamentally reshape financial analysis and strategic decision making. Furthermore, it is possible that its integration with emerging technologies such as artificial intelligence, blockchain and cryptocurrencies could generate significant synergies, leading to the creation of new financial products and innovative business models. Technical hurdles, such as the need for more stable and scalable systems, continue to be an area of intense research to achieve scalability and full practicality of this technology in any area of knowledge.

Recognizing that the emergence of quantum computing in finance raises important regulatory and ethical issues is important. Financial regulators are beginning to assess the implications of these technologies, especially in terms of information security and financial system stability. Concerns about the potential exacerbation of inequalities in financial markets, due to the competitive advantages it could confer, require continued attention from the financial community, policy makers and all interested partners in this area.

A prominent trend is the development of hybrid algorithms that combine classical and quantum components, enabling a gradual transition to quantum adoption in today's financial systems. Cybersecurity is also highlighted, as the emergence of more powerful quantum computers poses risks to existing encryption systems. Post-quantum cryptography thus becomes a priority for securing transactions and data.

In this context, financial institutions and regulators must proactively prepare for a quantum future. This includes investments in research and development, staff training and adaptation of business models. Although technical challenges remain, the pace of progress in quantum finance suggests a possible transition to this reality in the coming decades. Success will depend on collaboration among key players, the flexibility of institutions, and a shared commitment to responsible innovation that balances opportunity and risk.

The future of quantum finance is promising, and its realization will depend on our collective ability to harness its potential while carefully managing its challenges and broader implications.

References

1. Li, Q., Wu, H., Qian, W., Li, X., Zhu, Q., Yang, S.: Portfolio Optimization Based on Quantum HHL Algorithm. In: Sun, X., Zhang, X., Xia, Z., and Bertino, E. (eds.) Artif. Intell. and Security. pp. 90-99. Springer International Publishing, Cham (2022)
2. Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., Simonetto, A., Woerner, S., and Yndurain, E.: Quantum computing for finance: State-of-the-art and future prospects. IEEE Trans. on Quantum Eng. (2020) https://doi.org/10.1109/TQE.2020.3030314
3. Javed, A. R., Ahmed, W., Pandya, S., Maddikunta, P. K. R., and Alazab, M., Gadekallu, T. R.: A survey of explainable artificial intelligence for smart cities. Electron. (2023) https://doi.org/10.3390/electronics12041020
4. Thakkar, S., Kazdaghli, S.,Mathur, N. and Kerenidis, I., Ferreira-Martins, A. J., Brito, S.: Improved financial forecasting via quantum machine learning. Quantum Mach. Intell. (2000) https://doi.org/10.1007/s42484-024-00157-0
5. Orús, R., Mugel, S., Lizaso, E.: Quantum computing for finance: Overview and prospects. Rev. in Phys. (2019) https://doi.org/10.1016/j.revip.2019.100028
6. Herman, Dylan and Googin, Cody and Liu, Xiaoyuan and Sun, Yue and Galda, Alexey and Safro, Ilya and Pistoia, Marco and Alexeev, Y.: Quantum computing for finance. Nat. Rev. Phys. (2023) https://doi.org/10.1038/s42254-023-00603-1
7. Urrútia, G., Bonfill, X.: PRISMA declaration: A proposal to improve the publication of systematic reviews and meta-analyses; [Declaración PRISMA: una propuesta para mejorar la publicación de revisiones sistemáticas y metaanálisis]. Med. Clinica (2010) https://doi.org/10.1016/j.medcli.2010.01.015

8. Bouland, A., Fefferman, B., Landau, Z., Liu, Y.: Noise and the Frontier of Quantum Supremacy. 2021 IEEE 62nd Annu. Symp. on Found. of Comput. Sci. (FOCS) (2022) https://doi.org/10.1109/FOCS52979.2021.00127
9. Pistoia, M., and Ahmad, S. F., Ajagekar, A., Buts, A., Chakrabarti, S., Herman, D., Hu, S., Jena, A., Minssen, P., Niroula, P., Rattew, A., Sun, Y., Yalovetzky, R.: Quantum Machine Learning for Finance ICCAD Special Session Paper. In *2021 IEEE/ACM Int. Conf. On Comput. Aided Design (ICCAD)* (2021) https://doi.org/10.1109/ICCAD51958.2021.9643469
10. YesÃƒÅ'Ã‚Â§iltasÃƒÅ'Ã‚Â§, Ö.: The Black-Scholes equation in finance: Quantum mechanical approaches. Physica A: Stat. Mech. and its Appl. (2023) doi:https://doi.org/10.1016/j.physa.2023.128909
11. Arraut, I.: Gauge symmetries and the Higgs mechanism in Quantum Finance. EPL. (2023) https://doi.org/10.1209/0295-5075/acedce
12. Ukpabi, D., Karjaluoto, H., Bötticher, A., Nikiforova, A., Petrescu, D., Schindler, P., Valtenbergs, V., Lehmann, L.: Framework for understanding quantum computing use cases from a multidisciplinary perspective and future research directions. Futures. (2023) https://doi.org/10.1016/j.futures.2023.103277
13. Peral-García, D., Cruz-Benito, J., García-Peñalvo, F. J.: Systematic literature review: Quantum machine learning and its applications. Comput. Sci. Rev. (2024) https://doi.org/10.1016/j.cosrev.2024.100619
14. Gill, S. S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M., Buyya, R.: Quantum computing: A taxonomy, systematic review and future directions. J. Softw.: Pract and Exp. (2022) https://doi.org/10.1002/spe.3039
15. Upadhyay, S., Dhande, V., Mankodi, I.N., Bhatacharjee, R., Mishra, A., Banerjee, A., Venkataraman, R.: Unraveling the Effect of COVID-19 on the Selection of Optimal Portfolio Using Hybrid Quantum-Classical Algorithms. In: *2022 IEEE Int. Conf. on Quantum Comput. and Eng. (QCE).* (2022) https://doi.org/10.1109/QCE53715.2022.00154
16. Tripathy, S.S., Koul, N., Patel, H.: Comparative Study between Quantum and Classical Methods: Few Observations from Portfolio Optimization Problem. In: *2022 Int. Conf. on Trends in Quantum Comput. and Emerging Bus. Technologies (TQCEBT).* (2022) https://doi.org/10.1109/TQCEBT54229.2022.10041475
17. Grossi, M., Ibrahim, N., Radescu, V., Loredo, R., Voigt, K., Von Altrock, C., Rudnik, A.: Mixed Quantum-Classical Method for Fraud Detection With Quantum Feature Selection. IEEE Trans. on Quantum Eng. (2022) https://doi.org/10.1109/TQE.2022.3213474
18. Liu, X., Yuan, X., Zhang, R., Ye, N.: Risk assessment and regulation algorithm for financial technology platforms in smart city. Comput. Intell. and Neurosci. (2022) https://doi.org/10.1155/2022/9903364
19. Balakrishnan, D., Mariappan, U., Raghavendra, P.G.M., Reddy, P.K., Dinesh, R.L.N., Jabiulla, S.B.: Quantum Neural Network for Time Series Forecasting: Harnessing Quantum Computing's Potential in Predictive Modeling. In: *2023 2nd Int. Conf. on Futuristic Tech. (INCOFT).* (2023) https://doi.org/10.1109/INCOFT60753.2023.10425465
20. Wang, L., Lee, R.S.T.: Stock Index Return Volatility Forecast via Excitatory and Inhibitory Neuronal Synapse Unit with Modified MF-ADCCA. Fractal Fract. (2023) https://doi.org/10.3390/fractalfract7040292
21. Šćekić, M., Šćepanović, S., Mitrović, S.: Implementation of quantum machine learning algorithms: a literature review. In: *2022 11th Mediterranean Conf. on Embedded Comput. (MECO).* (2022) https://doi.org/10.1109/MECO55406.2022.9797184
22. Coccia, M., Roshani, S.: Path-Breaking Directions in Quantum Computing Technology: A Patent Analysis with Multiple Techniques. J. of the Knowl. Econ. (2024). https://doi.org/10.1007/s13132-024-01977-y
23. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: *Proc. 35th Annu. Symp. Found. Comput. Sci.* (1994) https://doi.org/10.1109/SFCS.1994.365700
24. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Proc. of the twenty-eighth Annu. ACM Symp. on Theory of Comput.* (1996) https://doi.org/10.1145/237814.237866

25. Babukhin, D.: Harrow-Hassidim-Lloyd algorithm without ancilla postselection. Phys. Rev. A. (2023) https://doi.org/10.1103/PhysRevA.107.042408
26. Swan, M., Witte, F., Dos Santos, R.P.: Quantum Inf. Sci. IEEE Internet Comput. (2022) https://doi.org/10.1109/MIC.2021.3132591
27. Huang, H.-L., Xu, X.-Y., Guo, C., Tian, G., Wei, S.-J., Sun, X., Bao, W.-S., Long, G.-L.: Near-term quantum computing techniques: Variational quantum algorithms, error mitigation, circuit compilation, benchmarking and classical simulation. Sci. China: Phys., Mech. and Astron. (2023) https://doi.org/10.1007/s11433-022-2057-y
28. de Pedro, L., París Murillo, R., López de Vergara, J.E., López-Buedo, S., Gómez-Arribas, F.J.: VaR Estimation with Quantum Computing Noise Correction Using Neural Networks. Math. (2023) https://doi.org/10.3390/math11204355
29. Hellstern, G.: Analysis of a hybrid quantum network for classification tasks. IET Quantum Commun. (2021) https://doi.org/10.1049/qtc2.12017
30. Buonaiuto, G., Gargiulo, F., De Pietro, G., Esposito, M., Pota, M.: Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. Sci. Rep. (2023) https://doi.org/10.1038/s41598-023-45392-w
31. Hegade, N.N., Chandarana, P., Paul, K., Chen, X., Albarrán-Arriagada, F., Solano, E.: Portfolio optimization with digitized counterdiabatic quantum algorithms. Phys. Rev. Res. (2022) https://doi.org/10.1103/PhysRevResearch.4.043204
32. Zhang, C., Sjarif, N.N.A., Ibrahim, R.: Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020-2022. Wiley Interdiscip. Rev.: Data Min. and Knowl. Discov. (2024) https://doi.org/10.1002/widm.1519
33. Lewin, M., Campani, C.H.: Constrained portfolio strategies in a regime-switching economy. Financ. Mark. Portf. Manag. (2023). https://doi.org/10.1007/s11408-022-00414-x
34. Markowitz, H.: Portfolio Selection. The J. of Financ. (1952) https://doi.org/10.2307/2975974
35. Catalano, F., Nasello, L., Guterding, D.: Quantum Computing Approach to Realistic ESG-Friendly Stock Portfolios. Risks. 12, (2024). https://doi.org/10.3390/risks12040066
36. Ramaiah, K., Soundarabai, P.B.: A constrained multi-period portfolio optimization model based on quantum-inspired optimization. Multimedia Tools Appl. (2024) https://doi.org/10.1007/s11042-024-18597-y
37. Lim, D., Rebentrost, P.: A quantum online portfolio optimization algorithm. Quantum Inf. Process. (2024) https://doi.org/10.1007/s11128-024-04256-6
38. Aguilera, E., de Jong, J., Phillipson, F., Taamallah, S., Vos, M.: Multi-Objective Portfolio Optimization Using a Quantum Annealer. Math. (2024) https://doi.org/10.3390/math12091291
39. Miyamoto, K., Kubo, K.: Pricing Multi-Asset Derivatives by Finite-Difference Method on a Quantum Computer. IEEE Trans. on Quantum Eng. (2022) https://doi.org/10.1109/TQE.2021.3128643
40. Miyamoto, K.: Quantum algorithm for calculating risk contributions in a credit portfolio. EPJ Quantum Technol. (2022) https://doi.org/10.1140/epjqt/s40507-022-00153-y
41. Zhao, T., Sun, C., Cohen, A., Stokes, J., Veerapaneni, S.: Quantum-inspired variational algorithms for partial differential equations: application to financial derivative pricing. Quantitative Financ. (2023) https://doi.org/10.1080/14697688.2023.2259954
42. Kaneko, K., Miyamoto, K., Takeda, N., Yoshino, K.: Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance. Quantum Inf. Process. (2021) https://doi.org/10.1007/s11128-021-03127-8
43. Alaminos, D., Salas, M.B., Partal-Ureña, A.: Hybrid ARMA-GARCH-Neural Networks for intraday strategy exploration in high-frequency trading. Pattern Recognit. (2024) https://doi.org/10.1016/j.patcog.2023.110139
44. Dri, E., Aita, A., Giusto, E., Ricossa, D., Corbelletto, D., Montrucchio, B., Ugoccioni, R.: A More General Quantum Credit Risk Analysis Framework. Entropy. (2023) https://doi.org/10.3390/e25040593
45. Wilkens, S., Moorhouse, J.: Quantum computing for financial risk measurement. Quantum Inf. Process. (2023). https://doi.org/10.1007/s11128-022-03777-2

46. Leclerc, L., Ortiz-Gutiérrez, L., Grijalva, S., Albrecht, B., Cline, J.R.K., Elfving, V.E., Signoles, A., Henriet, L., Del Bimbo, G., Sheikh, U.A., Shah, M., Andrea, L., Ishtiaq, F., Duarte, A., Mugel, S., Cáceres, I., Kurek, M., Orús, R., Seddik, A., Hammami, O., Isselnane, H., M'Tamon, D.: Financial risk management on a neutral atom quantum processor. Phys. Rev. Res. (2023) https://doi.org/10.1103/PhysRevResearch.5.043117
47. Alcazar, J., Leyton-Ortega, V., Perdomo-Ortiz, A.: Classical versus quantum models in machine learning: insights from a finance application. Mach. Learn.: Sci. Technol. (2020) https://doi.org/10.1088/2632-2153/ab9009
48. Fernández-Lorenzo, S., Porras, D., García-Ripoll, J.J.: Hybrid quantum-classical optimization with cardinality constraints and applications to finance. Quantum Sci. Technol. (2021) https://doi.org/10.1088/2058-9565/abf9af
49. Qu, D., Matwiejew, E., Wang, K., Wang, J., Xue, P.: Experimental implementation of quantum-walk-based portfolio optimization. Quantum Sci. and Technol. (2024) https://doi.org/10.1088/2058-9565/ad27e9
50. Bhasin, N.K., Kadyan, S., Santosh, K., HP, R., Changala, R., Bala, B.K.: Enhancing Quantum Machine Learning Algorithms for Optimized Financial Portfolio Management. In: *2024 Third Int. Conf. on Intell. Tech. in Control, Optim. and Signal Process. (INCOS).* (2024) https://doi.org/10.1109/INCOS59338.2024.10527612
51. Qiu, Y., Liu, R., Lee, R.S.T.: The design and implementation of a deep reinforcement learning and quantum finance theory-inspired portfolio investment management system. Expert Syst. Appl. (2024) https://doi.org/10.1016/j.eswa.2023.122243
52. Lin, R., Xing, Z., Ma, M., Lee, R.S.T.: Dynamic Portfolio Optimization via Augmented DDPG with Quantum Price Levels-Based Trading Strategy. In: *2023 Int. Joint Conf. on Neural Netw. (IJCNN).* (2023) https://doi.org/10.1109/IJCNN54540.2023.10191785
53. Portelinha, M., Campani, C.H., Roquete, R.: The impacts of cryptocurrencies in the performance of Brazilian stocks' portfolios. Economics Bulletin.7 (2021)
54. Gómez, A., Leitao, Á., Manzano, A., Musso, D., Nogueiras, M.R., Ordóñez, G., Vázquez, C.: A Survey on Quantum Computational Finance for Derivatives Pricing and VaR. Archives of Comput. Methods in Eng. (2022) https://doi.org/10.1007/s11831-022-09732-9
55. Martin, A., Candelas, B., Rodríguez-Rozas, Á., Martín-Guerrero, J.D., Chen, X., Lamata, L., Orús, R., Solano, E., Sanz, M.: Toward pricing financial derivatives with an IBM quantum computer. Phys. Rev. Res. (2021) https://doi.org/10.1103/PhysRevResearch.3.013167
56. Zhuang, X.-N., Chen, Z.-Y., Xue, C., Wu, Y.-C., Guo, G.-P.: Quantum Encoding and Analysis on Continuous Time Stochastic Process with Financial Applications. Quantum. (2023) https://doi.org/10.22331/q-2023-10-03-1127
57. Gigante, G., Zago, A.: DARQ technologies in the financial sector: artificial intelligence applications in personalized banking. Qualitative Res. in Financ. Markets. (2023) https://doi.org/10.1108/QRFM-02-2021-0025
58. Henge, S.K., Jayaraman, G., Sreedevi, M., Rajakumar, R., Rashid, M., Alshamrani, S.S., Alnfiai, M.M., Alghamdi, A.S.: Secure keys data distribution based user-storage-transit server authentication process model using mathematical post-quantum cryptography methodology. Netw. and Heterog. Media. (2023) https://doi.org/10.3934/NHM.2023057
59. Alaminos, D., Salas, M.B., Fernández-Gámez, M.A.: Quantum Computing and Deep Learning Methods for GDP Growth Forecasting. Comput. Economics. (2022) https://doi.org/10.1007/s10614-021-10110-z
60. Lu, J., Hu, Y., Li, X., Zhu, Q., Wu, H., Yang, S.: A New Stock Forecasting Model by Hidden Quantum Markov Models. Lect. Notes in Comput. Sci. (including subseries Lect. Notes in Artif. Intell. and Lect. Notes in Bioinformatics). (2022) https://doi.org/10.1007/978-3-031-06788-4_37
61. Rivera-Ruiz, M.A., Mendez-Vazquez, A., López-Romero, J.M.: Time Series Forecasting with Quantum Machine Learning Architectures. Lect. Notes in Computer Sci. (including subseries Lect. Notes in Artif. Intell. and Lect. Notes in Bioinformatics). (2022) https://doi.org/10.1007/978-3-031-19493-1_6

62. Alaminos, D., Esteban, I., Salas, M.B., Callejón, A.M.: Quantum neural networks for forecasting inflation dynamics. J. of Sci. and Ind. Res. (2020) https://doi.org/10.56042/jsir.v79i2.68439
63. Cao, Y., Zhou, X., Fei, X., Zhao, H., Liu, W., Zhao, J.: Linear-layer-enhanced quantum long short-term memory for carbon price forecasting. Quantum Mach. Intell. (2023) https://doi.org/10.1007/s42484-023-00115-2
64. Pushpak, S.N., Jain, S.: An Implementation of Quantum Machine Learning Technique to Determine Insurance Claim Fraud. In: *2022 10th Int. Conf. on Reliability, Infocom Tech. Optim. (Trends and Future Directions) (ICRITO).* (2022) https://doi.org/10.1109/ICRITO56286.2022.9964828
65. Shabbir, A., Shabir, M., Javed, A.R., Chakraborty, C., Rizwan, M.: Suspicious transaction detection in banking cyber-physical systems. Comput. & Electr. Eng. (2022) https://doi.org/10.1016/j.compeleceng.2021.107596
66. Innan, N., Sawaika, A., Dhor, A., Dutta, S., Thota, S., Gokal, H., Patel, N., Khan, M.A.-Z., Theodonis, I., Bennai, M.: Financial fraud detection using quantum graph neural networks. Quantum Mach. Intell. (2024) https://doi.org/10.1007/s42484-024-00143-6
67. Pena Tapia, E., Scarpa, G., Pozas-Kerstjens, A.: A didactic approach to quantum machine learning with a single qubit. Physica Scripta. 98, (2023). https://doi.org/10.1088/1402-4896/acc5b8
68. Said, D.: Quantum Computing and Machine Learning for Cybersecurity: Distributed Denial of Service (DDoS) Attack Detection on Smart Micro-Grid. Energies. (2023) https://doi.org/10.3390/en16083572
69. Kubo, K., Nakagawa, Y.O., Endo, S., Nagayama, S.: Variational quantum simulations of stochastic differential equations. Phys. Rev. A. (2021) https://doi.org/10.1103/PhysRevA.103.052425
70. Manjunath, C., Marimuthu, B., Ghosh, B.: Analysis of Nifty 50 index stock market trends using hybrid machine learning model in quantum finance. Int. J. of Electr. and Comput. Eng. (2023) https://doi.org/10.11591/ijece.v13i3.pp3549-3560
71. Harrigan, M.P., Sung, K.J., Neeley, M., Satzinger, K.J., Arute, F., Arya, K., Atalaya, J., Bardin, J.C., Barends, R., Boixo, S., Broughton, M., Buckley, B.B., Buell, D.A., Burkett, B., Bushnell, N., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Demura, S., Dunsworth, A., Eppens, D., Fowler, A., Foxen, B., Gidney, C., Giustina, M., Graff, R., Habegger, S., Ho, A., Hong, S., Huang, T., Ioffe, L.B., Isakov, S.V., Jeffrey, E., Jiang, Z., Jones, C., Kafri, D., Kechedzhi, K., Kelly, J., Kim, S., Klimov, P.V., Korotkov, A.N., Kostritsa, F., Landhuis, D., Laptev, P., Lindmark, M., Leib, M., Martin, O., Martinis, J.M., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Mohseni, M., Mruczkiewicz, W., Mutus, J., Naaman, O., Neill, C., Neukart, F., Niu, M.Y., O'Brien, T.E., O'Gorman, B., Ostby, E., Petukhov, A., Putterman, H., Quintana, C., Roushan, P., Rubin, N.C., Sank, D., Skolik, A., Smelyanskiy, V., Strain, D., Streif, M., Szalay, M., Vainsencher, A., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Zhou, L., Neven, H., Bacon, D., Lucero, E., Farhi, E., Babbush, R.: Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nat. Phys. (2021) https://doi.org/10.1038/s41567-020-01105-y
72. Radanliev, P.: Digital security by design. Security J. (2024). https://doi.org/10.1057/s41284-024-00435-3
73. Sotelo, R., Corbelletto, D., Dri, E., Giusto, E., Montrucchio, B.: Quantum Computing in Finance: The Intesa Sanpaolo Experience. IEEE Eng. Manag. Rev. (2024) https://doi.org/10.1109/EMR.2024.3373796
74. How, M.-L., Cheah, S.-M.: Business Renaissance: Opportunities and challenges at the dawn of the Quantum Computing Era. Bus. (2023) https://doi.org/10.3390/businesses3040036
75. Nikouei, M.A., Darvazeh, S.S., Amiri, M. Artificial Intelligence and Financial Markets in Smart Cities. In: Chakraborty, C., Lin, J.CW., Alazab, M. (eds) Data-Driven Mining, Leran. and Analytics for Secured Smart Cities. Adv. Sci. and Technol. for Secur. Appl. Springer, Cham. (2021)

76. Schetakis, N., Aghamalyan, D., Boguslavsky, M., Rees, A., Rakotomalala, M., Griffin, P.R.: Quantum machine Learning. for credit scoring. Math. (2024) https://doi.org/10.3390/math12091391
77. Grody, A.D.: Addressing cyber risk in financial institutions and in the financial system. J. of Risk Manag. in Financ. Institutions. 13, 155-16 (2020)
78. Dong, Y., Zheng, H., Zhu, J.: A narrative review on Quantum Finance Theory. Int. J. of Quantum Inf. (2024) https://doi.org/10.1142/S0219749924500163
79. Petticrew, M. and Roberts, H. (2001) Starting the Review: Refining the Question and Defining the Boundaries. In: Syst. Rev. in the Soc. Sci. 1st edn. Wiley, New York
80. Padilla V, García M (2006) Introducción a las finanzas. Compañía Editorial Continental, México

From Classical Challenges to Quantum Solutions: Grover's Algorithm for the N-Queens Problem

Oscar Montiel, Ulises Orozco-Rosas, Daniel López, and Moisés Sánchez

Abstract The N-Queens problem, a classic challenge in combinatorial optimization, requires placing N queens on an $N \times N$ chessboard such that no two queens threaten each other. This problem is NP-complete and # P-complete, indicating that classical solutions have exponential time complexity as N increases. This chapter explores the application of Grover's Algorithm, a quantum search algorithm known to provide quadratic speedup in unstructured search scenarios, to efficiently address the N-Queens problem. We present a comprehensive methodology for designing the quantum oracle necessary to enforce non-attacking constraints and detail the mathematical formulations for row, column, and diagonal checks. The implementation of this oracle and the associated quantum circuit is demonstrated using IBM Quantum Experience's simulator, showing a practical approach to leveraging quantum computational advantages. We analyze the simulation results for smaller board sizes and discuss the scalability challenges and hardware limitations of current quantum technology. The chapter highlights the potential of quantum algorithms to significantly outperform classical approaches, although practical implementations for larger N remain constrained by existing quantum hardware capabilities. Insights into optimizing quantum circuits and the integration of quantum-classical hybrid approaches are discussed as promising avenues for future research in quantum computing applications to complex combinatorial problems.

O. Montiel (✉) · D. López · M. Sánchez
Instituto Politécnico Nacional-CITEDI, Tijuana, B.C., Mexico
e-mail: oross@ipn.mx

D. López
e-mail: dlopez@citedi.mx

M. Sánchez
e-mail: msanchez@citedi.mx

U. Orozco-Rosas
CETYS Universidad, Tijuana, B.C., Mexico
e-mail: ulises.orozco@cetys.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_15

1 Introduction

The N-Queens problem, first posed by Max Bezzel in 1848, presents a classic combinatorial challenge: placing N queens on an $N \times N$ chessboard so that no two queens threaten each other. Although the problem is straightforward for smaller values of N, its complexity increases exponentially with board size, presenting significant computational challenges.

This exponential growth in complexity makes classical solutions impractical for large N, as they typically involve computationally intensive search strategies. The advent of quantum computing offers a novel paradigm for addressing such problems, leveraging quantum properties such as superposition and entanglement to perform computations in ways that classical computers cannot.

Among various quantum algorithms, Grover's Algorithm emerges as particularly promising due to its potential to provide quadratic speedup over classical unstructured search approaches [1]. This work explores the application of Grover's Algorithm to the N-Queens problem, focusing on the development of a suitable quantum oracle, evaluating the algorithm's performance against classical methods, and evaluating the scalability of this quantum approach using IBM's Quantum Experience simulator.

Through this study, our aim is not only to demonstrate the practical application of quantum computing to a well-known combinatorial problem but also to highlight the broader implications of quantum technologies in solving complex computational tasks. The insights gained here underscore the transformative potential of quantum algorithms, particularly in their ability to outperform classical computational methods for problems with exponential complexity growth.

1.1 Relevance and Computational Significance

The N-Queens problem, while historically intriguing, is of considerable importance in both theoretical and applied computing realms. It serves as a benchmark for developing and testing algorithmic strategies that extend beyond mere puzzle solving to applications requiring complex constraint satisfaction and optimization. Solutions to this problem are emblematic of broader challenges in fields ranging from cybersecurity to distributed systems [2, 3].

In the realm of artificial intelligence and optimization, the N-Queens problem is utilized as a standard testbed to evaluate algorithms under constrained conditions, pushing the limits of algorithmic design and efficiency [4].

From a theoretical point of view, the problem is a cornerstone in the study of NP-completeness and #P-completeness, serving as a prime example of how seemingly simple problems can expose the intrinsic complexities and limitations of classical computing approaches [5]. This dual classification emphasizes the difficulty: While verifying whether a given arrangement of queens is valid can be achieved in polynomial time, finding a solution or counting the total number of valid solutions poses a significant computational challenge. The complexity of classical solutions varies

greatly; heuristic methods can occasionally achieve polynomial-time complexity, while exhaustive search algorithms may exhibit factorial-time complexity, especially as N increases [6].

This backdrop sets the stage for the exploration of quantum computing solutions like Grover's Algorithm, which offers a theoretical quadratic speedup over classical unstructured search methods. By potentially reducing the search complexity for solutions, quantum approaches highlight a pivotal shift in computational paradigms, underscoring the need for innovations that could alter the landscape of algorithmic problem-solving. This exploration not only deepens our understanding of quantum computational advantages, but also catalyzes the development of quantum technologies that promise to tackle complex problems more efficiently than ever before.

1.2 Classical Solutions and Their Limitations

Historically, a variety of classical approaches have been developed to address the N-Queens problem, each with its unique strategies and associated challenges. These approaches range from brute force methods to more sophisticated algorithms, such as Backtracking, Branch-and-bound, and various heuristic and metaheuristic techniques, including Genetic Algorithms, Simulated Annealing, and local search [7, 8].

The brute force method, while simple and straightforward, it involves generating all possible configurations of queens on the board and checking each to determine if it meets the non-attacking condition. This method guarantees a solution but is computationally expensive with a time complexity that increases factorially with the number of queens ($N!$), rendering it impractical for large values of N [9].

Backtracking is a more refined approach that incrementally builds candidates for the solution and abandons a candidate ("backtracking") as soon as it determines that the candidate cannot possibly be completed to a valid solution. Although more efficient than brute force, backtracking is still subject to combinatorial explosion and can become computationally prohibitive as N increases, especially in the worst-case scenarios where many partial solutions must be evaluated [10].

Branch-and-bound and heuristic methods, such as simulated annealing and genetic algorithms, offer alternative strategies that can sometimes find solutions more quickly than backtracking by avoiding exhaustive enumeration of all possible configurations. However, these methods do not guarantee a solution and may not find the optimal solution in all cases. They are often used when the search space is too large to handle by exact methods but can still face challenges regarding convergence and computational overhead when scaling to larger boards [7, 11].

Metaheuristic techniques, which are high-level problem-independent algorithmic frameworks, provide strategies to guide the search process to explore large spaces of potential solutions. Although these approaches can be effective for many optimization problems, they require careful parameter tuning and may not consistently yield optimal solutions for all instances of the N-Queens problem [8, 12].

In summary, while classical algorithms for the N-Queens problem have been refined over the years, they inherently suffer from significant limitations due to the

combinatorial nature of the problem. These limitations manifest themselves as an increase in computational resources and time with the growth of N, limiting their practical applicability to larger instances. This situation underscores the ongoing need for innovative approaches, potentially based on quantum computing, to overcome the barriers faced by classical methods.

1.3 The Quantum Advantage with Grover's Algorithm

Quantum computing introduces a significant shift in computational paradigms by leveraging quantum mechanical phenomena such as superposition and entanglement, which allow for the processing of information in fundamentally different ways compared to classical computing. Among the diverse array of quantum algorithms, Grover's Algorithm stands out for its potential to dramatically accelerate unstructured search problems. Developed by Lov Grover in 1996, this algorithm improves search efficiency by achieving a quadratic speedup, reducing the time complexity from $O(N)$ for classical methods to $O(\sqrt{N})$ for quantum processes [1].

The essence of Grover's Algorithm lies in its use of quantum superposition to evaluate multiple possibilities simultaneously and quantum interference to amplify the probability amplitudes of desirable states. It starts by preparing a superposition of all possible states in the quantum register. A quantum oracle, a specific quantum operation, then inverts the amplitudes of states that meet the search criteria. Subsequently, a diffusion operator (Grover operator) enhances the amplitudes of these marked states while reducing those of the unmarked ones, thereby increasing the probability of measuring a desired state upon completion of the algorithm [13].

Despite its theoretical advantages, practical implementation of Grover's Algorithm is challenged by the current limitations of quantum technology. Effective deployment of this algorithm requires a quantum oracle that can efficiently encode problem constraints into a quantum framework, along with a quantum computing environment that maintains coherence and minimizes operational errors throughout the process [14]. Physical realization in contemporary quantum computers is hindered by issues such as short qubit coherence times and high error rates in quantum gates, which can significantly diminish the potential quadratic speedup [15].

The future of Grover's Algorithm appears highly promising, driven by ongoing advancements in quantum hardware, error correction methodologies, and algorithmic optimization techniques. These developments are poised to significantly expand its practical applicability, not only for the N-Queens problem but also for a wide spectrum of combinatorial complex challenges that remain intractable for classical computing. As quantum technologies continue to mature, Grover's Algorithm is anticipated to emerge as a cornerstone in the quantum computing toolkit, enabling efficient resolution of intricate computational tasks with unparalleled effectiveness [13, 15].

Overall, while Grover's Algorithm offers a significant theoretical improvement in the efficiency of search tasks, bridging the gap between its theoretical potential and practical application remains a critical area of research in the quantum computing field [16].

1.4 Chapter Organization

The chapter is structured as follows: Sect. 2 provides the theoretical foundations necessary to understand Grover's algorithm. This section begins with the mathematical definition of the unstructured search problem and continues with an overview of the fundamental principles of quantum computing. It concludes with a detailed explanation of the algorithm's mathematical structure and operational process, offering the essential background for its application. Section 3 is dedicated to solving the N-Queens problem using Grover's Algorithm. In this section, we describe how to model the constraints of the N-Queens problem mathematically and explain their integration into the quantum framework. Section 4 presents a case study that applies Grover's Algorithm to the 5-Queens problem. It includes an explanation of the algorithm, covering the initialization of the quantum circuit, the design of the oracle, and the application of the diffusion operator to amplify the probability of correct solutions. Additionally, this section provides an analysis of the iterative dynamics of Grover's Algorithm for the 5-Queens Problem, demonstrating its feasibility and highlighting its execution process. Section 5 focuses on the experimental setup and the results obtained from the implementation. This section analyzes the algorithm's performance, scalability, and computational efficiency. Comparisons with classical approaches are also provided to highlight the advantages and limitations of the quantum approach. Section 6 concludes the chapter by summarizing the key findings of the study and emphasizing their relevance to quantum computing research. Furthermore, it proposes future directions for research, such as improving the oracle design, refining error correction techniques, and extending the algorithm's application to other combinatorial problems.

2 Theoretical Foundations

This section presents the mathematical and theoretical foundations underlying Grover's algorithm, focusing on its application to the N-Queens problem. We begin by formalizing the general search problem within the context of quantum computing and proceed to detail Grover's algorithm, highlighting its potential for providing a quadratic speedup over classical search methods.

2.1 Mathematical Representation of the Search Problem

The unstructured search problem can be mathematically defined as follows. Given a black-box function $f : \{0, 1\}^N \to \{0, 1\}$, where $N = 2^n$ represents the size of the search space, and n is the number of qubits required to encode the search space, the objective is to find an input $x \in \{0, 1\}^N$ such that $f(x) = 1$. This function $f(x)$

outputs 1 exclusively for the index $x = \omega$ that satisfies the search condition. In classical computing, the location of ω typically requires $O(N)$ evaluations of $f(x)$, making it computationally expensive as N grows. Grover's algorithm, on the other hand, leverages quantum parallelism and amplitude amplification to reduce the number of evaluations to $O(\sqrt{N})$, achieving a significant improvement in search efficiency [1].

2.2 Quantum Computing Fundamentals

Quantum computing represents a paradigm shift in computational science, leveraging the principles of quantum mechanics to process information in ways that are fundamentally different from classical computing. At its core lies the qubit, the quantum analogue of the classical bit. Unlike a classical bit, which is restricted to binary states 0 or 1, a qubit can exist in a superposition of states, mathematically expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where α and β are complex probability amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$. This property enables quantum computers to explore multiple possibilities simultaneously.

Quantum systems also exhibit entanglement, a phenomenon in which qubits become interconnected such that the state of one qubit cannot be described independently of the state of the others. Entanglement is critical for many quantum algorithms, including Grover's algorithm, as it facilitates the global manipulation of quantum states.

An n-qubit quantum register can represent 2^n states simultaneously, described as:

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x\,|x\rangle\,, \tag{2}$$

where α_x represents the amplitude of each basis state $|x\rangle$. This exponential scalability is one of the key advantages of quantum computing over classical approaches.

2.3 Grover's Algorithm: Mathematical Foundations and Operational Steps

Grover's algorithm leverages quantum superposition and amplitude amplification to provide a quadratic speedup for unstructured search problems. By reducing the complexity from $O(N)$ to $O(\sqrt{N})$, where N is the total number of items, it significantly improves the efficiency of searches through large unsorted databases. Entanglement may emerge during the algorithm's execution, particularly when implementing the

oracle or the diffusion operator, but it is not fundamentally required as a computational resource.

The algorithm operates through a sequence of oracle and diffusion operator applications, iteratively amplifying the amplitudes of valid solution states. The key steps are outlined in Algorithm 1, which provides a structured representation of Grover's search process [17].

Algorithm 1 Grover's Algorithm for Quantum Search

1: **Input:** N possible states and an oracle U_ω that marks valid solutions.
2: **Output:** A solution state $|x\rangle$ satisfying the problem constraints.
3: **Step 1: Initialization:** Apply Hadamard gates to all qubits to prepare the uniform superposition state:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle . \tag{3}$$

4: **Step 2: Oracle Application:** Apply the oracle U_ω to mark valid solutions by flipping their phase:

$$U_\omega |x\rangle = \begin{cases} -|x\rangle , & \text{if } x \text{ is a solution,} \\ |x\rangle , & \text{otherwise.} \end{cases} \tag{4}$$

5: **Step 3: Diffusion Operator:** Apply the diffusion operator D to amplify the amplitudes of the marked states (see Fig. 1):

$$D = 2 |s\rangle \langle s| - I. \tag{5}$$

This operator reflects the quantum state about the mean amplitude of all computational basis states, effectively amplifying the probability of measuring valid solutions. In practical implementations, D is constructed using Hadamard gates as $D = H^{\otimes n}(2 |0\rangle \langle 0| - I)H^{\otimes n}$, which is mathematically equivalent to the reflection over $|s\rangle$, since $|s\rangle = H^{\otimes n} |0\rangle$.
6: **Step 4: Iterative Process:** Repeat Steps 2 and 3 for approximately:

$$k \approx \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rfloor \text{ iterations.} \tag{6}$$

7: **Step 5: Measurement:** Measure the quantum register to obtain a solution state $|x\rangle$ with high probability.

The iterative process of the Grover Algorithm can be mathematically represented as:

$$|\psi\rangle \to U_\omega D |\psi\rangle \to (U_\omega D)^2 |\psi\rangle \to \cdots \to (U_\omega D)^k |\psi\rangle .$$

The oracle U_ω applies a phase inversion to the marked states, as defined in Eq. (4). Then, the diffusion operator D is applied using Eq. (5), which amplifies the amplitudes of the marked states; in this equation, the state $|s\rangle$ was obtained for the first time with Eq. (3) and represents the uniform superposition state; this state will evolve throughout approximately k iterations. As a result, the probability of measuring a valid solution becomes significantly high.

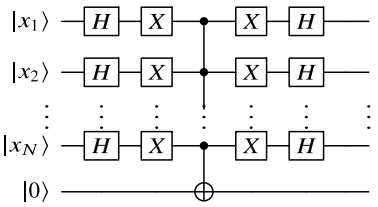


Fig. 1 Schematic of Grover's diffusion operator for the N-Queens problem. The circuit applies Hadamard gates, X-gates, and a multi-controlled NOT gate with an ancilla qubit initialized in $|0\rangle$. The diffusion operator amplifies the amplitudes of valid solutions by reflecting the states about the average amplitude

The formal presentation of Grover's Algorithm highlights the structured interplay between the oracle and the diffusion operator. The algorithm begins by preparing the quantum register in a uniform superposition state, ensuring equal probability amplitudes for all candidate solutions.

The oracle U_ω then identifies the solution states by inverting their phase, marking them as valid configurations. Following oracle application, the diffusion operator D amplifies the probability amplitudes of the marked states. As illustrated in Fig. 1, the diffusion operator reflects the quantum states about their average amplitude, which progressively increases the likelihood of observing valid solutions during measurement.

The iterative combination of the oracle and diffusion operator gradually brings the quantum state closer to the valid solution states. The approximate number of iterations needed to maximize the probability of observing a valid solution is provided in Eq. (6); where M represents the number of valid solutions.

Grover's Algorithm highlights the efficiency of quantum computing for unstructured search problems. By alternating the oracle operator and the diffusion operator, the algorithm progressively amplifies the amplitudes of valid solutions, increasing their probability of identification. As illustrated in Fig. 1, the diffusion operator reflects quantum states about the mean amplitude, concentrating the probability on valid configurations. This iterative process makes Grover's Algorithm a key approach for quantum search techniques, including optimization problems such as the N-Queens problem.

The implementation of Grover's diffusion operator D may or may not require an ancilla qubit, depending on the theoretical or practical context of the quantum circuit.

As shown in Fig. 2, the diffusion operator can theoretically be constructed using only Hadamard gates, X gates, and a multi-controlled Z gate. The primary role of D is to reflect the quantum states in their average amplitude, increasing the probabilities of valid solutions. In this idealized version, no ancilla qubit is needed because the multi-controlled Z gate directly applies the required phase inversion to the $|0\rangle$ state. This is valid in a scenario where the multi-controlled gate can be implemented perfectly and without hardware limitations.

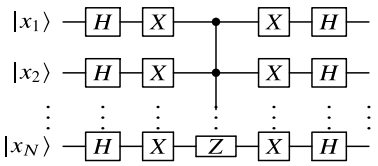


Fig. 2 Schematic of Grover's diffusion operator for the N-Queens problem without ancilla qubits. The circuit applies Hadamard gates, X-gates, and a multi-controlled Z gate, which performs the phase inversion directly on the $|0\rangle$ state. This idealized implementation assumes a direct multi-controlled phase flip operation

However, in practical implementations on real quantum hardware, such as Noisy Intermediate-Scale Quantum (NISQ) devices, the absence of a direct multi-controlled Z gate introduces challenges:

- Multi-controlled gates are not natively available and must be decomposed into basic gates, such as CNOT gates and T gates.
- During this decomposition, auxiliary qubits (ancilla) are often required to reduce the circuit depth and facilitate the proper execution of multi-controlled operations.

Therefore, the need for an ancilla qubit depends on the context:

- In **theoretical implementations**, as depicted in Fig. 2, the diffusion operator does not require ancilla qubits. The phase inversion is applied directly using an ideal multi-controlled Z gate.
- In **practical implementations**, where gate decompositions are necessary, ancilla qubits become essential to manage the complexity of the circuit and ensure correct execution on hardware.

Therefore, while Grover's diffusion operator can theoretically omit ancilla qubits, practical realizations often rely on them to account for hardware constraints. Figure 2 illustrates the simplified theoretical version of the diffusion operator, where phase inversion is applied without ancillary qubits.

It is important to note that the operator $D = 2|s\rangle\langle s| - I$ is equivalent to $D = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$, where $|s\rangle = H^{\otimes n}|0\rangle$. This equivalence is critical for implementing the diffuser using standard quantum gates, as direct reflection about $|s\rangle$ is not natively available in hardware [17].

3 Solving the N-Queens Problem Using Grover's Algorithm

The adaptation of Grover's Algorithm for the N-Queens problem offers a compelling demonstration of the potential of quantum computing in solving combinatorial optimization problems efficiently. This section outlines the design and implementation of the quantum oracle U_ω, which is crucial for encoding the problem's constraints into Grover's search framework.

3.1 Encoding the Chessboard: Qubit Representation

To implement the N-Queens problem within a quantum framework, it is essential to encode the chessboard layout into qubits effectively. Each position on an $N \times N$ chessboard can be uniquely represented by a qubit state, where y_{ij} is a binary variable indicating whether a queen occupies position (i, j) on the board. Therefore, the quantum register $|\Psi\rangle$ is defined as a collection of N^2 qubits, each qubit representing a potential queen position:

$$|\Psi\rangle = \bigotimes_{i=1}^{N} \bigotimes_{j=1}^{N} \left|y_{ij}\right\rangle, \tag{7}$$

where $y_{ij} \in \{0, 1\}$. Here, $\left|y_{ij}\right\rangle = 1$ indicates that a queen is placed at position (i, j), while $\left|y_{ij}\right\rangle = 0$ represents an empty square.

3.2 Formulating the Oracle for Validating Non-attacking Configurations

The oracle's primary role in Grover's Algorithm is to evaluate whether a queen configuration on the chessboard satisfies the non-attacking condition by checking the constraints on rows, columns, and diagonals. Table 1 summarizes the mathematical model for the N-Queens problem, describing these constraints.

The objective is to identify a valid configuration in which N queens are placed on the chessboard so that no two queens threaten each other. The constraints ensure:

- Each row contains exactly one queen.
- Each column contains exactly one queen.
- No two queens share the same right diagonal, characterized by a constant value of $i + j$.
- No two queens share the same left diagonal, characterized by a constant value of $i - j$.

In the context of Grover's Algorithm, these constraints are encoded within the oracle as quantum operations. The oracle marks valid configurations by inverting their phase, effectively identifying solutions that meet all constraints. This encoding process plays a pivotal role in ensuring the efficiency and correctness of the quantum algorithm.

These conditions can be encoded to enable Grover's algorithm to operate efficiently within the constraints of the N-Queens problem. Encoding involves setting up the oracle circuit to recognize and mark the states that represent invalid queen placements. This marking process is essential, as it enables the search procedure to suppress invalid configurations and focus amplitude amplification exclusively on valid ones. As a result, the search space is significantly reduced, and the number of iterations required to find a solution is minimized.

Table 1 Mathematical model of the N-Queens problem

Set:	
	B: The chessboard of size $N \times N$
Parameters:	
	N: Number of rows and columns on the chessboard B
Variables:	
	$y_{ij} = \begin{cases} 1 & \text{If a queen is placed at position } (i, j) \text{ on } B, \\ 0 & \text{Otherwise} \end{cases}$
Objective:	
	Find a configuration of N queens on B such that all constraints are satisfied.
Constraints:	
	$\sum_{j=1}^{N} y_{ij} = 1 \quad \forall i \in \{1, \ldots, N\}$ (Only one queen per row)
	$\sum_{i=1}^{N} y_{ij} = 1 \quad \forall j \in \{1, \ldots, N\}$ (Exactly one queen per column)
	$\sum_{\substack{i,j \\ i-j=k}} y_{ij} \leq 1 \quad \forall k \in \{-N+1, \ldots, N-1\}$ (Left diagonals: $i - j$ constant)
	$\sum_{\substack{i,j \\ i+j=k}} y_{ij} \leq 1 \quad \forall k \in \{2, \ldots, 2N\}$ (Right diagonals: $i + j$ constant)

4 Case Study: Applying Grover's Algorithm to the 5-Queens Problem

This study investigates the application of quantum computing, specifically Grover's Algorithm, to efficiently solve the 5-Queens problem, a specialized instance of the N-Queens puzzle. The objective is to position five queens on a 5×5 chessboard such that no two queens threaten each other, in accordance with the constraints of the puzzle.

The approach consists of initializing qubits in superposition, employing an oracle to enforce the constraints of the 5-Queens problem, and utilizing Grover's diffusion operator to amplify the probabilities of valid configurations. Each step in this quantum workflow is elaborated upon in the subsequent sections.

In this case study, for simplification, we begin with one queen's position already known, placed at the upper left corner of the chessboard, as shown in Fig. 3. This setup reduces the overall complexity and helps us to explain how we find the solution more clearly. It allows us to focus exclusively on locating the positions of the remaining queens, showing how Grover's Algorithm can be applied effectively. Although it is possible to tackle the problem with all initial positions of the queens unknown, this would greatly increase the complexity because it requires exploring a larger range of possible configurations and interactions among the queens. However, setting the starting position of one queen simplifies the demonstration of the methodology.

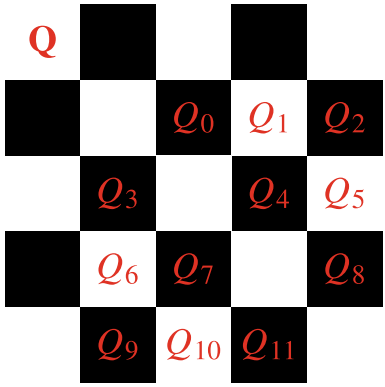


Fig. 3 A 5×5 chessboard, featuring a fixed queen Q in the upper-left corner. The white squares represent prohibited positions, as they are threatened by the fixed queen. The viable positions for placing additional queens are labeled as Q_0 to Q_{11}, which correspond to the candidate qubits used in the quantum circuit for solving the 5-Queens problem

Based on this configuration, we determine the number of qubits required for the quantum representation by considering the viable positions for queens and the ancillary qubits necessary for verifying constraints and phase inversion.

Total Ancilla Qubits for Constraints and Diffusion: The total number of qubits is calculated as:

$$\begin{aligned}\textbf{Total Qubits} &= 12\ (\text{State Qubits}) + 4\ (\text{Row Ancillas}) + 4\ (\text{Column Ancillas}) \\ &+17\ (\text{Diagonal Ancillas}) + 18\ (\text{Oracle Phase Inversion Ancilla}) = 55\ \text{qubits}\end{aligned}$$

Thus, this study requires a total of 55 qubits: 12 for representing the states (viable queen positions) and 18 for verifying constraints and applying phase inversion. However, it is possible to reduce the total number of used qubits by recycling some intermediate ancillary qubits and achieving an implementation with less than 50 qubits.

4.1 Initialization Phase in Quantum Circuit for Grover's Algorithm

The initialization phase is critical in preparing the quantum circuit to solve the 5-Queens problem, as it sets the stage for the entire quantum computation. This phase involves setting all qubits in a superposition state, enabling the simultaneous exploration of all possible queen configurations on the 5×5 chessboard. The initialization procedure is described as follows.

- **Qubit Preparation**: Each qubit, representing a potential position for a queen on the chessboard, is initialized to the zero state, denoted by $|0\rangle$.
- **Superposition**: A Hadamard gate (H-gate) is applied to each qubit to create a superposition of states. This gate transforms the state $|0\rangle$ to $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$, effectively enabling each qubit to represent both the presence and absence of a queen in that position.

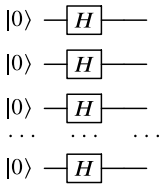


Fig. 4 Simplified representation of the initialization stage for the quantum register. Each qubit is initialized in a superposition state $|+\rangle$ by applying a Hadamard gate to the $|0\rangle$ state, illustrating the uniform distribution of amplitudes across all possible computational basis states

The initialization in superposition allows the quantum algorithm to consider multiple configurations per qubit, significantly reducing the quantum resources required and potentially lowering the complexity of the problem compared to a direct one-to-one qubit-to-chess position mapping. This compact encoding ensures that the quantum register can represent a broader set of valid configurations. Subsequent gates in the oracle and diffusion steps will entangle these qubits, which is essential for maintaining the interdependencies of the queen positions according to the N-Queens rules.

Figure 4 shows a simplified representation of the initialization stage adapted for the actual number of qubits used.

4.2 Oracle Design

The oracle in Grover's algorithm for the 5-Queens problem is designed to evaluate the legality of each configuration of queens represented by the quantum state of the qubits. It uses a combination of quantum gates to check all constraints related to rows, columns, and diagonals.

Row and Column Constraints:

Row and Column Constraints are implemented to ensure that each row and column contains exactly one queen.

- **Row Constraints**: These are implemented to ensure that each row has exactly one queen. The constraints are detailed in the matrix form and specific equations that enforce one queen per selected rows:

$$\begin{aligned} Q_0 + Q_1 + Q_2 &= 1 \\ Q_3 + Q_4 + Q_5 &= 1 \\ Q_6 + Q_7 + Q_8 &= 1 \\ Q_9 + Q_{10} + Q_{11} &= 1 \end{aligned} \tag{8}$$

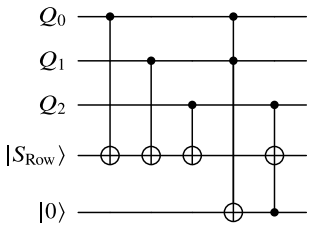


Fig. 5 Quantum gate configuration for enforcing the row constraint $Q_0 + Q_1 + Q_2 = 1$, ensuring that exactly one queen is placed in the corresponding row of the chessboard. The circuit utilizes a combination of controlled-NOT (CNOT) gates and Toffoli (multi-controlled) gates to evaluate the validity of the quantum state. The first three CNOT gates, controlled by the qubits Q_0, Q_1, and Q_2, compute the parity (i.e., the XOR) of these three qubits and store it in the auxiliary qubit $|S_{\text{Row}}\rangle$. This operation sets $|S_{\text{Row}}\rangle = Q_0 \oplus Q_1 \oplus Q_2$, which equals 1 not only when exactly one qubit is in state $|1\rangle$, but also when all three are in $|1\rangle$, leading to a potential false positive. To eliminate this ambiguity, the circuit incorporates an additional ancilla qubit initialized in $|0\rangle$, which interacts with the primary ancilla $|S_{\text{Row}}\rangle$ through a series of Toffoli gates. These gates detect configurations with more than one active qubit (i.e., more than one queen) and flip the state of $|S_{\text{Row}}\rangle$ accordingly, canceling the false positive that arises from the XOR operation alone. In this way, the circuit correctly marks the configuration as valid (with $|S_{\text{Row}}\rangle = 1$) if and only if exactly one of the qubits Q_0, Q_1, or Q_2 is in the $|1\rangle$ state. All other configurations, including those with zero or multiple queens in the same row, result in $|S_{\text{Row}}\rangle = 0$, indicating a violation of the constraint. This logic ensures Grover's oracle can effectively distinguish and mark only valid placements for amplitude amplification

The row constraints ensure that each row on the chessboard can host no more than one queen. This is critical to avoid any direct horizontal attacks between queens. The enforcement of these constraints within the quantum circuit is depicted in Fig. 5, which illustrates the specific setup of quantum gates used to check this condition.

In the context of Grover's algorithm, these gates function by flipping the state of an auxiliary qubit if more than one qubit in a given row is found in the state representing the presence of a queen (typically the $|1\rangle$ state). This auxiliary qubit becomes an integral part of the oracle's mechanism to mark configurations that violate the chess rules, allowing the algorithm to effectively ignore or downplay these during the amplitude amplification process.

Column Constraints:

Similar logic applies to columns, ensuring one queen per column through quantum gates that check the sum of qubits representing columns. Figure 6 shows the quantum circuit implementation of the column constraint $Q_3 + Q_6 + Q_9 = 1$.

$$\begin{aligned} Q_3 + Q_6 + Q_9 &= 1 \\ Q_0 + Q_7 + Q_{10} &= 1 \\ Q_1 + Q_4 + Q_{11} &= 1 \\ Q_2 + Q_5 + Q_8 &= 1 \end{aligned} \tag{9}$$

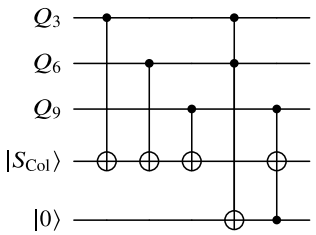


Fig. 6 Quantum gate configuration for enforcing the column constraint $Q_3 + Q_6 + Q_9 = 1$, ensuring that exactly one queen is placed in the corresponding column of the chessboard. This circuit mirrors the structure used for the row constraints: it first computes the parity of the input qubits using a sequence of CNOT gates and then applies Toffoli gates for correction, detecting and negating invalid configurations with more than one active qubit. The ancilla qubit $|S_{\text{Col}}\rangle$ remains in state $|1\rangle$ only if exactly one of Q_3, Q_6, or Q_9 is in state $|1\rangle$, thereby validating the constraint

Diagonal Constraints:

Diagonal constraints prevent more than one queen from sharing the same diagonal, classified into positive and negative diagonals.

- **Positive Diagonals**: Constraints prevent more than one queen from sharing positive diagonals. Figure 7 shows the circuits to implement these five constraints.

$$\begin{aligned} Q_3 + Q_0 &< 2 \\ Q_6 + Q_1 &< 2 \\ Q_9 + Q_7 + Q_4 + Q_2 &< 2 \\ Q_{10} + Q_5 &< 2 \\ Q_{11} + Q_8 &< 2 \end{aligned} \tag{10}$$

- **Negative Diagonals**: Similar restrictions apply for negative diagonals. Figure 8 shows the circuit to implement these four constraints.

$$\begin{aligned} Q_6 + Q_{10} &< 2 \\ Q_3 + Q_7 + Q_{11} &< 2 \\ Q_0 + Q_4 + Q_8 &< 2 \\ Q_1 + Q_5 &< 2 \end{aligned} \tag{11}$$

Verification of All Constraints and Phase Inversion

In the quest to solve the 5-Queens problem using quantum computing, a fundamental step involves the construction of a quantum circuit that can verify all necessary constraints simultaneously. These constraints include ensuring that no two queens threaten each other along any row, column, or diagonal. The goal of this circuit

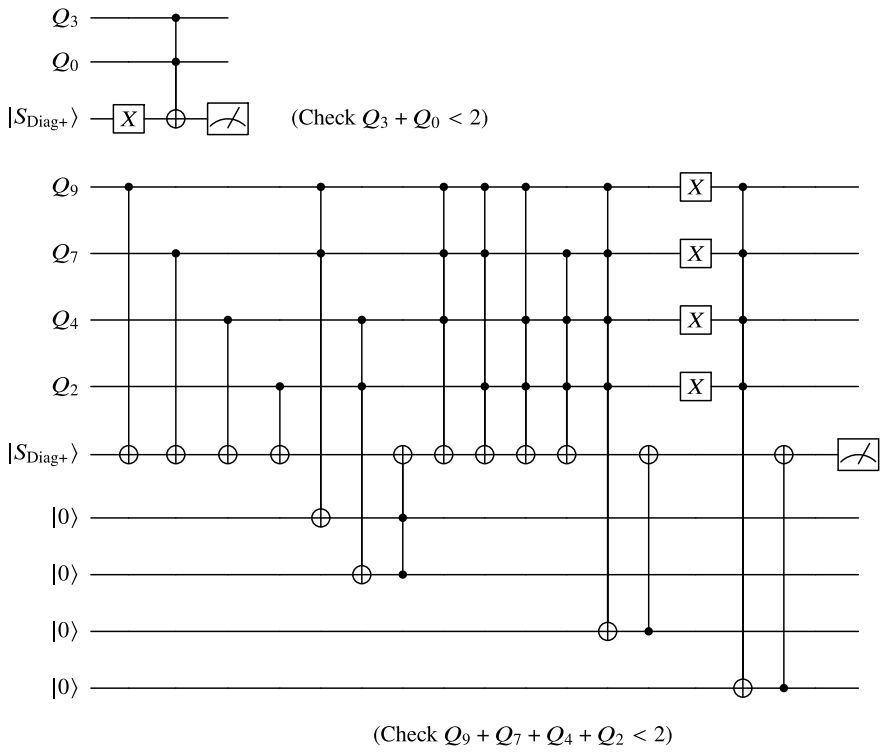


Fig. 7 Quantum circuits for verifying positive diagonal constraints in the 5-Queens problem. Each circuit ensures that no more than one queen is placed along any positive diagonal, preventing mutual attacks. For constraints involving two qubits;e.g., $Q_3 + Q_0 < 2$ or $Q_6 + Q_1 < 2$, controlled gates flip a unique ancilla qubit when both qubits are in the $|1\rangle$ state, indicating a violation. For multi-qubit constraints (e.g., $Q_9 + Q_7 + Q_4 + Q_2 < 2$), a sequence of controlled gates accumulates the results into a distinct ancilla qubit for that specific constraint. Each ancilla qubit is measured independently at the end of the operation to confirm whether the constraint is satisfied, where $|0\rangle$ denotes no violation and $|1\rangle$ denotes a violation. The use of separate ancilla qubits for each constraint ensures that the circuits can operate independently, enforcing the positive diagonal constraints and maintaining the integrity of potential solutions in the problem's solution space. This constraint is validated when $\left|S_{Diag+}\right\rangle = |1\rangle$

is to validate that a given configuration of queens on the board satisfies all these conditions, thereby identifying valid solutions.

To achieve this, the quantum circuit incorporates not only the logical checks for all constraints but also a phase-inversion operation. The phase inversion step is a core component in Grover's algorithm, marking the valid configurations by flipping the phase of states that satisfy all constraints. This operation ensures that the amplitudes of the valid solutions are enhanced during the iterative process of the algorithm, increasing the likelihood of measuring these solutions at the end.

The construction of this verification circuit requires combining the circuits for row, column, and diagonal constraints with unique ancilla qubits for each type of

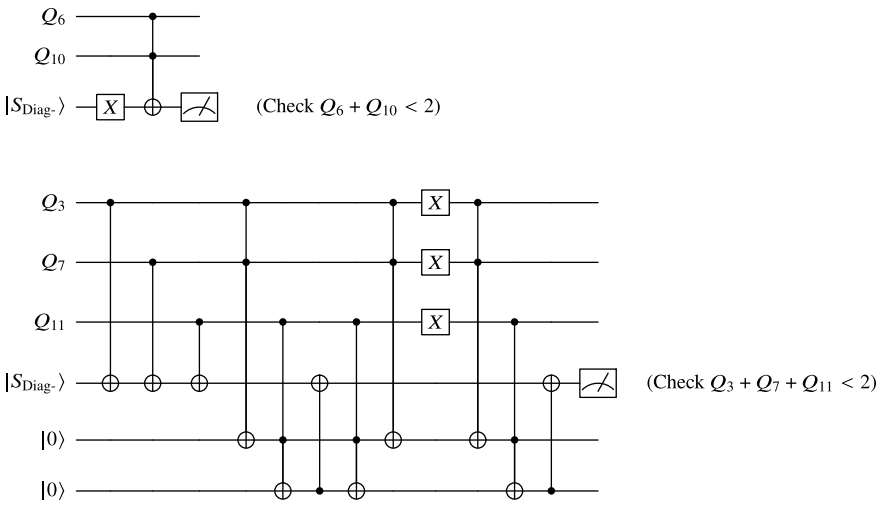


Fig. 8 Quantum circuits for verifying negative diagonal constraints in the 5-Queens problem. Each circuit ensures that no more than one queen is placed along any negative diagonal, preventing mutual attacks. For constraints involving two qubits (e.g., $Q_6 + Q_{10} < 2$), a controlled-NOT gate is used to flip a unique ancilla qubit when both qubits are in the $|1\rangle$ state. For constraints involving three qubits (e.g., $Q_3 + Q_7 + Q_{11} < 2$), a series of controlled gates accumulates the results into a distinct ancilla qubit. The ancilla qubits are measured independently to indicate whether the constraints are satisfied, where $|0\rangle$ denotes no violation and $|1\rangle$ denotes a violation. The use of separate ancilla qubits for each constraint ensures independent verification and collectively enforces the integrity of the negative diagonal constraints in the problem's solution space. This constraint is validated when $\left|S_{Diag-}\right\rangle = |1\rangle$

check. After verifying the constraints, the phase inversion operation is applied to the states that pass all checks, completing the process of constraint validation in the quantum computation framework for the 5-Queens problem.

Figure 9 shows the verification circuit and the phase inversion operation.

4.2.1 Preparation for the Grover's Diffuser

To effectively utilize Grover's diffusion operator, it is necessary to return to a clean state where the ancilla qubits used during the verification and phase inversion steps are reset to their initial $|0\rangle$ state. This ensures that the input to the diffusion operator consists solely of the state qubits representing the valid configurations of queens, without any residual entanglement or information from the ancilla qubits.

Starting from the circuit obtained during the verification and phase inversion stage, we must traverse the circuit in reverse order, systematically undoing the operations applied to the ancilla qubits. This reversal process, known as uncomputation, is crucial to eliminate any entanglement introduced during constraint verification. This process includes:

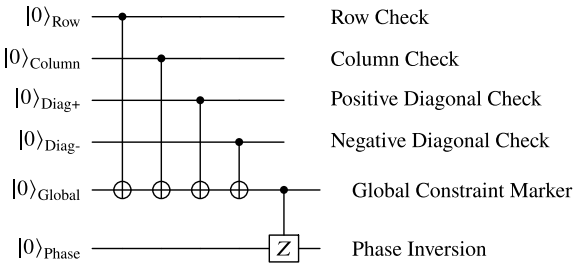


Fig. 9 Simplified quantum circuit representation for the verification of all constraints and phase inversion in the 5-Queens problem. The inputs to this circuit correspond to the ancilla qubits associated with the row, column, positive diagonal, and negative diagonal constraints. Each ancilla qubit records the result of its respective constraint check. A global ancilla qubit is used to combine the results of all constraint checks, ensuring that the phase inversion operation is only applied if all constraints are satisfied simultaneously. If any constraint fails, the global ancilla prevents the multi-controlled Z gate from acting on the phase inversion qubit. This simplified representation ensures that only valid solutions are marked for amplitude amplification in Grover's algorithm. Note that the Toffoli gate to validate the answer is missing in this circuit since adding it will increase the circuit size

- **Resetting Ancilla States**: To ensure that the ancilla qubits do not interfere with subsequent operations, we reverse the logic gates previously applied during the verification of row, column, and diagonal constraints. By reversing these operations, the ancilla qubits are returned to the ground state $|0\rangle$, effectively removing any residual quantum information.
- **Maintaining Phase Inversion Effects**: While the phase inversion operation targets the state qubits and does not alter the ancilla qubits directly, its effect on the global state is preserved. This ensures that the configurations deemed valid continue to be emphasized in the subsequent steps of the algorithm, particularly during the application of Grover's diffuser.
- **Preserving State Qubits**: The state qubits, which represent the potential configurations of queens on the chessboard, remain unchanged after the oracle and phase inversion steps. It is crucial to maintain their states, since these qubits serve as inputs to Grover's diffusion operator, which amplifies the probabilities of valid configurations.

Reversing the operations after completing the verification circuit specifically resets the ancilla qubits, ensuring that they return to their initial states. This step is essential for preserving the marked superposition of the state qubits generated by the oracle phase inversion. By ensuring that the ancillary qubits are reset, the quantum system avoids undesired interference, preserving the integrity of the computational process. This amplification is essential for optimizing the probability of successfully observing these valid states on measurement.

By resetting the ancillary qubits and preserving the state of the state-qubits, this step forms a crucial bridge between the verification and diffusion stages of Grover's algorithm, facilitating a seamless transition within the quantum computational work-

flow. This preparation ensures that Grover's algorithm operates with maximum efficiency, minimizing error propagation and optimizing the overall solution to the 5-Queens problem.

4.3 Grover Diffusion Operator for the 5-Queens Problem

Figure 10 presents a quantum circuit diagram illustrating the application of the diffusion operator for the 5-Queens problem, enclosed within a box labeled "Diffuser". This symbolic representation abstracts the internal sequence of quantum gates that compose the diffusion operator, emphasizing its functional role within the overall algorithm. The specific components are described as follows:

- **Qubits $|q_0\rangle$ to $|q_4\rangle$**: These qubits represent the positions of the five queens on a chessboard. Each qubit is configured to reflect a specific potential location, which is essential for encoding and manipulating candidate solutions to the problem.
- **Diffuser Box**: This box simplifies the depiction of the Grover diffusion operator's function in the circuit. It symbolizes the collective operations—a series of Hadamard gates, phase shifts, and another sequence of Hadamard gates—applied to all qubits simultaneously. This transformation reflects the inversion about the mean mechanism, which lies at the core of Grover's amplification process.

The optimal number of iterations for the Grover diffusion operator in the 5-Queens problem is calculated to be approximately 13.88. This value, when applying the floor function, is reduced to 13 iterations, as detailed in Eq. (6).

4.4 Implementation and Iterative Dynamics of Grover's Algorithm for the 5-Queens Problem

Grover's algorithm utilizes an iterative application of an oracle and a diffusion operator to amplify the amplitudes of states that represent valid solutions. The number of

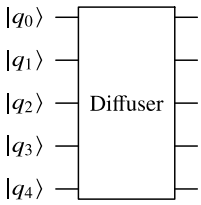


Fig. 10 Quantum circuit diagram illustrating the Grover Diffusion Operator encapsulated within the "Diffuser" box, highlighting the collective manipulation of qubits representing the 5-Queens problem. The ancilla qubits are not shown. Note that ancillary qubits, which are reset prior to this step, are not shown in the diagram

iterations required for the diffuser is given by the formula in Eq. (6), where N is the total number of possible configurations and M is the number of valid solutions. For the 5-Queens problem, accurately determining N and M without extensive simulations or deep combinatorial analysis is non-trivial. This is because N encompasses all potential states of the quantum system, while M includes only those states that meet the constraints of the problem. Therefore, these values are typically estimated based on theoretical approximations or empirical observations from preliminary experiments, reflecting the intricate and dynamic nature of solving quantum problems. The result of a theoretical formula might not be an integer; therefore, it is recommended to round this number to the nearest lower integer (via the floor function) and adjust the algorithm by testing slightly fewer or slightly more iterations. This fine-tuning is crucial, as overamplification due to too many iterations can lead to a decrease in the probabilities of observing valid solutions, counteracting the intended effect of the algorithm.

Grover's algorithm ideally amplifies the probability of observing a valid state. However, if the system is measured immediately after applying the calculated iterations, it usually collapses to one of the valid solutions rather than sampling uniformly across all valid solutions. To obtain multiple valid answers in problems where there is more than one solution (as in the 5-Queens problem), the following strategies may be employed:

- **Restart and Repetition**: After performing a measurement and recording a solution, the system can be restarted and the algorithm can be executed several times. This technique takes advantage of the probabilistic nature of the algorithm to explore different solutions in each run.
- **Oracle Modification**: Modify the oracle to exclude previously found solutions. This can be achieved by marking already discovered configurations as invalid, thus forcing the algorithm to focus on unexplored valid states.
- **Statistical Analysis**: Execute the algorithm several times and perform a statistical analysis of the results obtained to identify frequently observed configurations, as likely valid solutions.

These strategies require careful planning and can increase the total computational cost, but are effective in systematically exploring the solution space.

This integrated approach, combining precision in the choice of the number of iterations with methods for exploring multiple solutions, maximizes the effectiveness of Grover's algorithm in solving complex problems like the 5-Queens in a quantum context.

The designed quantum circuit integrates separate sections, each responsible for verifying a specific set of constraints: Row Constraints, Column Constraints, Positive Diagonal Constraints, and Negative Diagonal Constraints. Additionally, the circuit includes a component that simultaneously verifies that all constraints are met and that phase inversion is correctly implemented; Fig. 11 illustrates these concepts.

Each section uses a series of quantum gates to control ancillary qubit, depending on the state of the qubits that represent the positions of the queens, as explained in Sect. 4.2. If any part of the circuit detects that the constraints are not met, for example,

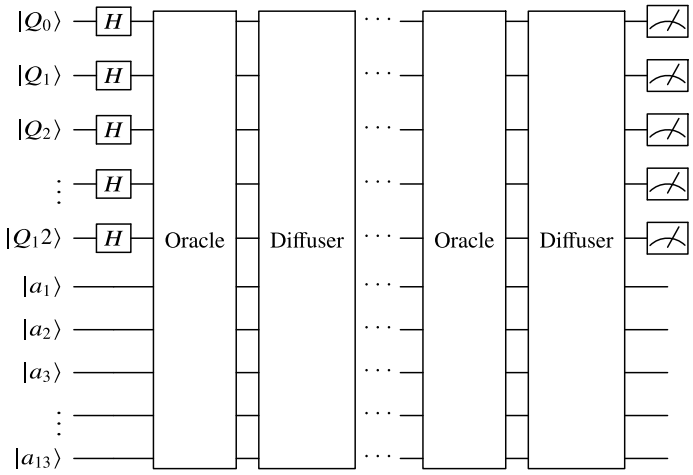


Fig. 11 Complete quantum circuit for the 5-Queens problem using Grover's Algorithm. The circuit includes multiple iterations of the oracle and diffuser operations, represented as black boxes. The circuit structure reflects the repeated application of the oracle followed by the diffusion operator, as prescribed by Grover's iterative framework. Ancillary qubits ($|a_1\rangle, |a_2\rangle, \ldots, |a_{13}\rangle$) are utilized to verify the correct execution of the oracle and to ensure that the solution satisfies all placement constraints

if there is more than one queen per row, column, or diagonal, the corresponding ancilla qubit will flip its state to show this violation.

The ancillas associated with each set of constraints play a crucial role in the integrated verification process, serving as indicators of whether the queens' placements adhere to the rules or violate them:

- If all ancillas remain in their initial state (typically $|0\rangle$) following the circuit operation, this confirms that the configuration violates the queens' placement rules.
- Conversely, if any ancilla is found in the state $|1\rangle$, it indicates that the queens' placement configuration is valid.

5 Experiments and Results

The primary goal of this experiment is to evaluate and compare the performance of classical algorithms–specifically Backtracking, Branch-and-bound, and Linear Programming–against the simulated quantum algorithm to solve the N-Queens problem. This comparison aims to highlight the potential computational advantages of quantum approaches by assessing execution times and computational complexities for each method across various problem sizes, from 4 to 17 queens. Furthermore, the experiment explores the theoretical implications of scaling the problem up to 128 queens.

5.1 Experiment Setup

Algorithms tested include Backtracking, Branch-and-bound, Linear Programming, and the Hybrid Quantum Algorithm. The classical algorithms were executed for queen configurations ranging from 4 to 17 on a server equipped with a Xeon Gold 5218 @ 2.3 GHz and 512 GB RAM, running Linux Ubuntu 20.04.2 LTS. The 5-Queens problem was experimentally tested using the Hybrid Quantum Algorithm on the IBM Quantum Experience platform [18, 19].

5.2 Results

The classical algorithms and the quantum algorithm were evaluated based on their estimated performance and theoretical computational complexity. Because the quantum algorithm was simulated on classical hardware, execution time measurements are excluded, as they do not provide an accurate reflection of quantum computational performance. Instead, the quantum algorithm is analyzed based on its circuit depth, qubit requirements, and scalability potential.

This subsection details the performance of three classical algorithms – Backtracking, Branch-and-bound, and Linear Programming – as well as the Hybrid Quantum Algorithm. Each algorithm was selected because of its distinct approach to solving the N-Queens problem, providing a comprehensive basis for comparing their computational efficiency and scalability across different configurations. The inclusion of the quantum algorithm allows for a direct comparison of the classical and quantum computational capabilities.

Backtracking is a recursive, depth-first search algorithm that explores all possible configurations on the board, backtracking when a configuration violates the problem's constraints. Branch-and-bound improves upon the naive backtracking approach by pruning paths that cannot possibly lead to a valid solution, thus potentially reducing the search space significantly. Linear Programming, on the other hand, formulates the problem as a set of linear inequalities and attempts to find solutions through optimization techniques.

Each algorithm was executed on the specified server, ensuring that performance metrics are comparable and reliable. Table 2 summarizes the execution times measured in different sizes of the N-Queens problem, from 4 to 17 queens [7]. These metrics provide insights into each algorithm's scalability and efficiency as the problem size increases, particularly highlighting the best execution times for each queen size in bold, typically found with Branch-and-bound. Subsequently, this method was used as the basis for comparing the computational complexities shown in Table 3. The same table presents the computed complexities of each algorithm based on their algorithmic design, highlighting the quantum advantage. Additionally, the column "Iterations vs Repetitions ratio" quantifies the efficiency of the quantum algorithm in reducing the number of necessary operational cycles compared to classical

Table 2 Execution times for classical algorithms and the estimated execution time for the hybrid quantum algorithm, with a speedup factor comparison. For each number of queens, the best classical execution time is highlighted in bold, and the corresponding speedup factor for the quantum algorithm is provided, demonstrating the quantum advantage in computational efficiency

Number of queens	Execution time (s)			Execution time (s)	Speedup
	Backtracking	Branch-and-bound	Linear programming	Hybrid quantum	Factor
4	0.00007	**0.0004**	0.006335	10^{-9}	4×10^5
5	0.0002	**0.0005**	0.01257	10^{-9}	5×10^5
6	0.0008	**0.0009**	0.01768	10^{-9}	9×10^5
7	0.003405	**0.00243**	0.035475	10^{-9}	2.43×10^6
8	0.015385	**0.008565**	0.11619	10^{-9}	8.565×10^6
9	0.074975	**0.03553**	0.524145	10^{-9}	3.553×10^7
10	0.22796	**0.104655**	2.00735	10^{-9}	1.0466×10^8
11	1.199305	**0.471255**	11.46305	10^{-9}	4.7126×10^8
12	6.877155	**2.548845**	63.38689	10^{-9}	2.5488×10^9
13	42.3444	**14.530725**	534.99516	10^{-9}	1.4531×10^{10}
14	278.702635	**89.19125**	–	10^{-9}	8.9191×10^{10}
15	1,896.905275	**583.959185**	–	10^{-9}	5.83956×10^{11}
16	13,887.82939	**4,115.949215**	–	10^{-9}	4.1159×10^{12}
17	109,411.7678	**30,531.33199**	–	10^{-9}	3.0531×10^{13}

Table 3 Comparative computational complexities for classical and hybrid quantum algorithms, highlighting the consistent complexities of Backtracking and Branch-and-bound as $O(2^N)$, compared against the factorial growth in Linear Programming and the more efficient polynomial growth of the Hybrid Quantum algorithm. The complexities marked in bold for Branch-and-bound indicate the basis for comparison with the Hybrid Quantum algorithm in each scenario. Additionally, the "Iterations vs Repetitions Ratio" column quantifies the efficiency of the quantum algorithm in reducing the number of necessary operational cycles compared to classical approaches

Number of queens	Backtracking	Branch-and-bound	Linear programming	Hybrid quantum	Iterations versus repetitions ratio
4	$O(2^4)$	$\mathbf{O(2^4)}$	$O(4!)$	$O(2)$	16 : 4
5	$O(2^5)$	$\mathbf{O(2^5)}$	$O(5!)$	$O(2.23)$	32 : 5
8	$O(2^8)$	$\mathbf{O(2^8)}$	$O(8!)$	$O(2.83)$	256 : 8
16	$O(2^{16})$	$\mathbf{O(2^{16})}$	$O(16!)$	$O(4)$	65, 536 : 16
32	$O(2^{32})$	$\mathbf{O(2^{32})}$	$O(32!)$	$O(5.66)$	4.29×10^9 : 32
64	$O(2^{64})$	$\mathbf{O(2^{64})}$	$O(64!)$	$O(8)$	1.84×10^{19} : 64
128	$O(2^{128})$	$\mathbf{O(2^{128})}$	$O(128!)$	$O(11.31)$	3.40×10^{38} : 128
256	$O(2^{256})$	$\mathbf{O(2^{256})}$	$O(256!)$	$O(16)$	1.16×10^{77} : 256
512	$O(2^{512})$	$\mathbf{O(2^{512})}$	$O(512!)$	$O(22.63)$	1.34×10^{154} : 512

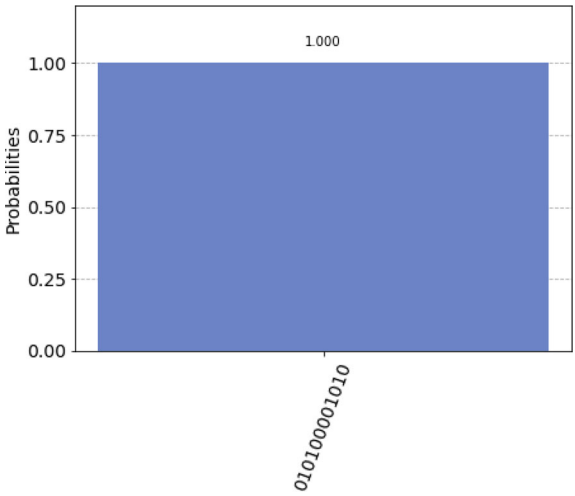


Fig. 12 Histogram of the output states from Grover's algorithm indicating the frequency of valid configurations

approaches, with Branch-and-bound complexities highlighted in bold, indicating the selected efficiency over Backtracking in each scenario, serving as a comparison basis against the quantum algorithm.

Upon successful application of Grover's algorithm to the 5-Queens problem, the results were visualized in the histogram shown in Fig. 12, which displays the distribution of valid solutions. The histogram reveals that the solution 010100001010 appeared to be the most probable configuration. This binary string represents the placement of queens on the chessboard in such a way that no two queens attack each other.

In this solution, the binary string 010100001010 is read from left to right, where Q_0 corresponds to the most significant bit, and Q_{11} corresponds to the least significant bit. The placement of queens is as follows: A queen is placed at positions Q_1, Q_3, Q_8, and Q_{10}; contrary to this, no queen is placed at positions Q_0, Q_2, Q_4, Q_5, Q_6, Q_7, Q_9, and Q_{11}. Each bit in the binary string corresponds to a potential position for a queen on the chessboard, based on the original diagram that defines valid states; i.e., Fig. 3.

The mapping ensures that all queens are positioned according to the rules of the N-Queens problem: no two queens share the same row, column, or diagonal. This correct and precise arrangement, as visualized in Fig. 13, demonstrates that no queen threatens another.

The solution obtained by the quantum algorithm proves the efficiency of this quantum approach. It highlights how quantum computing can successfully solve complex combinatorial problems. The clear visual representation also shows the power of quantum computation to find optimal solutions even in contexts where classical methods encounter exponential growth in complexity.

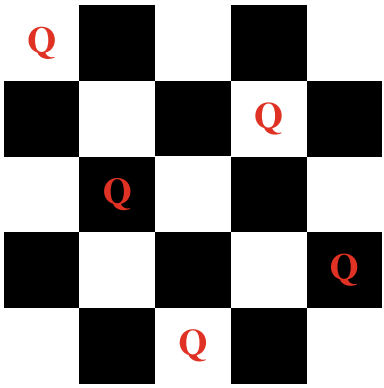


Fig. 13 Optimal placement of queens on a 5×5 chessboard as determined by Grover's Algorithm, illustrating a successful quantum computation solution. The configuration avoids any queens threatening each other, with a fixed queen in the upper-left corner (Row 0, Column 0). The diagram highlights the effective use of quantum states to solve the 5-Queens problem, showcasing the potential of quantum computing in addressing complex combinatorial challenges

6 Conclusions

The implementation of Grover's Algorithm in the N-Queens problem highlights its notable advantage in searching within an unstructured solution space through quantum superposition. This approach contrasts sharply with classical algorithms, which require sequential testing of each possible input. By employing quantum superposition, Grover's Algorithm evaluates all possible inputs simultaneously, optimizing the search process significantly.

Initially, the algorithm uniformly superposes all qubits, assigning equal probability amplitude to every potential state. This uniform distribution is critical, as it ensures that no particular state is favored before the application of the oracle. The oracle then selectively alters the phase of states representing valid solutions, effectively distinguishing these states within the quantum superposition without necessitating distinct inputs for each evaluation.

The oracle function is pivotal, marking valid states by phase inversion, a process that underscores the dynamics of quantum mechanics. Following this, the Grover diffuser rebalances the amplitudes of all states, enhancing those marked by the oracle and diminishing others. This amplitude adjustment is repeated; each iteration is designed to progressively increase the probability amplitude of the correct solutions, optimizing the likelihood of their measurement.

The iterative application of the diffusion operator in Grover's Algorithm is crucial for enhancing the probability distribution across the quantum state space. This doesn't involve altering the initial inputs, which remain as a uniform superposition of all possible states. Instead, the role of the diffusion operator is to repeatedly increase the probabilities of states marked by the oracle as potential solutions. This improves our

chances of correctly identifying the solution when we measure the system. Through this method, the algorithm optimizes the search within the existing quantum states without modifying the input. The optimal number of iterations, approximately proportional to $\sqrt{N}$, maximizes the chances that the algorithm converges to a valid solution upon measurement. This process highlights the efficiency of the algorithm, leveraging quantum mechanics to reduce the computational complexity faced by its classical counterparts.

Grover's Algorithm is not static; it can be adapted based on known properties of the solution space or constraints, potentially enhancing its efficiency. For multisolution problems like the N-queens, modifying the oracle to exclude already found solutions allows for a systematic exploration of the solution space without restarting the algorithm, demonstrating its scalability and adaptability.

The practical implementation of this algorithm on the IBM Quantum simulator and the subsequent visualization of the results through histograms not only validate the theoretical model but also provide a tangible representation of the effectiveness of the algorithm. This practical demonstration confirms that quantum computing can be a viable tool for addressing combinatorial optimization problems, even when limited to current quantum hardware capabilities.

Moreover, the detailed design and operational description of the incorporated oracle provide a robust framework ready for integration into broader academic discussions or presentations.

Future research will focus on extending the application of Grover's Algorithm to other combinatorial optimization problems, including more complex variations of the N-Queens problem, such as the N-Queens Completion problem. This will explore scalability and efficiency in larger quantum computing environments. Additionally, efforts will be directed toward enhancing the quantum oracle design to improve algorithm performance and reduce computational overhead. Advanced quantum error correction techniques will also be investigated to address challenges posed by qubit decoherence and gate fidelity, aiming to achieve more reliable and scalable quantum computation.

References

1. L. K. Grover, "A fast quantum mechanical algorithm for database search," 1996. [Online]. Available: arXiv:quant-ph/9605043
2. J. Bell and B. Stevens, "A survey of known results and research areas for n-queens," *Discrete Mathematics*, vol. 309, no. 1, pp. 1–31, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0012365X07010394
3. A. Al-Herz, "Solving the n-queens problem for applications in vlsi circuit testing," *The International Arab Journal of Information Technology*, vol. 11, no. 6, pp. 542–549, 2014.
4. O. Moghimi and A. Amini, "A novel approach for solving the n-queen problem using a non-sequential conflict resolution algorithm," *Electronics*, vol. 13, no. 20, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/20/4065
5. I. P. Gent, C. Jefferson, and P. Nightingale, "Complexity of n-queens completion," *Journal of Artificial Intelligence Research*, vol. 59, p. 815 – 848, 2017.

6. S. G. S, P. Joshi, A. Barui, and P. K. Panigrahi, "A quantum approach to solve n-queens problem," 2023. [Online]. Available: arXiv:2312.16312
7. A. Arteaga, U. Orozco-Rosas, O. Montiel, and O. Castillo, *Evaluation and Comparison of Brute-Force Search and Constrained Optimization Algorithms to Solve the N-Queens Problem*. Cham: Springer International Publishing, 2022, pp. 121–140.
8. V. M. Saffarzadeh, P. Jafarzadeh, and M. Mazloom, "A hybrid approach using particle swarm optimization and simulated annealing for n-queen problem," *World Academy of Science, Engineering and Technology*, vol. 43, p. 974 – 978, 2010.
9. A. Alhassan, "Build and conquer: Solving n queens problem using iterative compression," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2019, pp. 1–5.
10. S. Güldal, V. Baugh, and S. Allehaibi, "N-queens solving algorithm by sets and backtracking," in *SoutheastCon 2016*, 2016, pp. 1–8.
11. S. A. Hussein and I. A. Zahid, "Improved naked mole-rat algorithm based on variable neighborhood search for the n-queens problem," *Iraqi Journal of Science*, vol. 65, no. 1, p. 528 – 545, 2024.
12. O. K. Majeed, R. Hashim Ali, A. Z. Ijaz, N. Ali, U. Arshad, M. Imad, S. Nabi, J. Tahir, and M. Saleem, "Performance comparison of genetic algorithms with traditional search techniques on the n-queen problem," in *2023 International Conference on IT and Industrial Technologies, ICIT 2023*, 2023.
13. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
14. R. Seidel, C. K.-U. Becker, S. Bock, N. Tcholtchev, I.-D. Gheorghe-Pop, and M. Hauswirth, "Automatic generation of grover quantum oracles for arbitrary data structures," *Quantum Science and Technology*, vol. 8, no. 2, p. 025003, jan 2023. [Online]. Available: https://doi.org/10.1088/2058-9565/acaf9d
15. J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018, https://doi.org/10.22331/q-2018-08-06-79.
16. Y. Sun and L.-A. Wu, "Quantum search algorithm on weighted databases," 2024. [Online]. Available: https://arxiv.org/abs/2312.01590. arXiv:2312.01590
17. J. Watrous, "Grover's algorithm," 2024, accessed: 2024-12-17. [Online]. Available: https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/grovers-algorithm
18. IBM, "Start quantum programming with ibm quantum experience," 2024, accessed: 2024-12-14. [Online]. Available: https://mediacenter.ibm.com/media/Start+quantum+programming+with+IBM+Quantum+Experience/1_qsxcanqp
19. B. Burd, *Quantum Computing Algorithms: Discover how a little math goes a long way*. Packt Publishing, 2021.

Cross-Platform Analysis of Quantum Circuit Simulators with State Collapse Mechanisms and Back-Action Tracking in C and Python

Paul O. Onoja, Seyi Tope Ogunji, Moises Sanchez-Adame, Oscar Montiel, Ushnik Nath, Lubna M Siddiqui, and Fernando Alonso Morquecho Lara

Abstract In this work we present an empirical-based comparative analysis of quantum circuit simulators that are implemented using C and Python programming languages, focusing on advanced measurement capabilities with state collapse mechanisms. The simulators support full quantum state evolution and feature our unique contribution that allows both projective (collapsing) and weak (non-collapsing) measurements. Our implementation extends traditional quantum simulation by incorporating flexible measurement schemes that can maintain quantum superposition when desired, enabling the study of quantum trajectories and partial measurements. The C implementation focuses on performance through optimized memory management and efficient tensor contractions. The Python version, while slower, offers better extensibility and seamless integration with existing scientific computing libraries and insightful contributions from the vast community of the language developers. This will facilitate fast prototyping and analysis thereby making the study of quantum simulation more accessible to researchers in the field. Both versions implement

P. O. Onoja, S.T.Ogunji, M. Sanchez-Adame, O. Montiel, F. Alonso Morquecho Lara, U. Nath, L. M. Siddiqui—Equal contribution.

P. O. Onoja
Department of Computer Science, Nile University of Nigeria, Abuja, FCT, Nigeria

S. T. Ogunji · M. Sanchez-Adame · O. Montiel (✉) · F. Alonso Morquecho Lara
CITEDI, Instituto Politécnico Nacional, Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

S. T. Ogunji
e-mail: seyitope@citedi.mx

M. Sanchez-Adame
e-mail: mosancheza@ipn.mx

F. Alonso Morquecho Lara
e-mail: fmorquecho@citedi.mx

U. Nath
Guru Gobind Singh Indraprastha University, Dwarka, New Delhi, Delhi, India

L. M. Siddiqui
Amity University, Noida, Uttar Pradesh, India

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_16

identical quantum operations, allowing validation of results across implementations. Our implementation achieves simulation capabilities of up to 29 qubits in C and 20 qubits in Python, with demonstrated zero back-action across weak measurements (0.1–0.5 strength). The direct method shows approximately 30% better performance than the tensor method for larger systems, while maintaining measurement precision with outcome distributions centered at 0.5. Validation through GHZ state preparation demonstrates perfect qubit correlations, confirming reliable quantum state manipulation. Performance benchmarks demonstrate the trade-offs between computational efficiency and development flexibility, providing insights for choosing appropriate tools for different quantum simulation scenarios.

Keywords Quantum circuit simulation · State collapse mechanisms · Weak measurements · Cross-platform implementation · Back-action tracking · GHZ state preparation · Tensor contractions · Performance optimization

1 Introduction

The quantum state representation in our implementation is built upon core data structures. The primary structure, StateVector, encapsulates the essential components of quantum state representation which are the number of qubits in the system (`num_qubits`), the state evolution [1] represented as

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle \tag{1}$$

where $U(t)$ represents the unitary time evolution operator, and its corresponding density matrix representation

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{2}$$

where p_i represents the classical probability of finding the system in state $|\psi_i\rangle$, the state space dimension calculated as $2^{\text{num\_qubits}}$, the selected simulation method, and an array storing quantum state amplitudes. For quantum operations, state normalization [1] is essential to maintain quantum mechanical constraints, defined as:

$$|\psi_{\text{norm}}\rangle = \frac{|\psi\rangle}{\sqrt{\sum_i |\psi_i|^2}} \tag{3}$$

where:

- $|\psi_{\text{norm}}\rangle$ is the normalized state vector
- $|\psi\rangle$ is the unnormalized state vector
- $|\psi_i|^2$ represents probability amplitude for each basis state.

The normalization condition ensures $\sum_i |\psi_i|^2 = 1$ is maintained throughout quantum operations. Additionally, a QuantumGate structure implements a 2×2 matrix representation for quantum operations.

In our implementation, we showcase three unique simulation methods that are implemented with both C and Python programming languages to provide room for a realistic benchmarking framework. The tensor contraction method (tensor_contract) employs bit manipulation for state indices and memory-efficient contraction operations. The direct state manipulation method (direct_apply_gate) performs in-place state vector updates with direct amplitude modifications. The matrix multiplication method (matrix_apply_gate) utilizes full matrix representation, though it is limited to systems of 14 qubits due to computing overhead constraints.

For quantum operations, we implemented four fundamental gates: the Hadamard gate (get_hadamard_gate) for creating superposition states, the X gate (get_x_gate) for quantum NOT operations, the T gate (get_t_gate) for phase rotations, and the CNOT gate (apply_cnot_gate) for two-qubit controlled-NOT operations. The fidelity of our gate operations [1] is measured by:

$$F_{\text{gate}} = \frac{1}{d^2}|\text{Tr}(U_{\text{ideal}}^{\dagger} U_{\text{actual}})|^2 \tag{4}$$

where:

- F_{gate} is the gate fidelity
- d is the dimension of the Hilbert space
- U_{ideal} is the ideal gate operation
- U_{actual} is the implemented gate operation.

The measurement system implements two types of measurements. Projective measurements utilize calculate_measurement_probability for computing measurement probabilities [2] defined as

$$P(\text{outcome}|\text{qubit}) = \sum_{i:(i\&(1\ll\text{qubit}))==\text{outcome}} |\psi_i|^2 \tag{5}$$

where the summation is over all basis states satisfying the bitwise condition for the measured qubit, and collapse_state for implementing state vector collapse. Weak measurements [3], mathematically expressed as

$$O_w = P_0 + (1 - s)(r - 0.5)\Delta \tag{6}$$

where:

- O_w is the weak measurement outcome
- P_0 is the probability of measuring state $|0\rangle$
- s is the measurement strength

- r is a random number between 0 and 1
- Δ is the measurement resolution parameter.

are implemented through perform_weak_measurement where we varied the measurement strength to study the effect of disturbance across the measurement regime. The weak measurement operator [2] is mathematically expressed as:

$$M(s) = \sqrt{1 + s(2P_0 - 1)}|0\rangle\langle 0| + \sqrt{1 - s(2P_0 - 1)}|1\rangle\langle 1| \tag{7}$$

where:

- $M(s)$ is the weak measurement operator
- s is the measurement strength parameter
- P_0 is the probability of measuring state $|0\rangle$.

The weak measurement is cleverly included in the implementation with the intent of providing non-collapsing measurement capabilities with a configurable strength parameter and back-action tracking.

Finally, data collection and analysis, which are very integral to our implementation are intensively incorporated to the simulation design. Runtime analysis is performed through measure_circuit_runtime for execution time measurement. The implementation includes basic validation through two key functions: validate_single_qubit for checking qubit index validity and validate_qubit_indices for validating two-qubit operation indices. These validation mechanisms ensure the integrity of quantum operations within the implementation.

The remaining part of this work is divided as follows: Sect. 2 details the robust methodological approach used for the implementation, Sect. 3 addresses the experimental results obtained and their comprehensive and rigorous analysis, Sect. 4 focuses on the conclusion drawn from the work and the future direction on further research work relating to the implementation.

2 Methodology

System Set-Up

The system used for this quantum simulation, marked as Microsoft WSL Virtual GPU (Device 008e) is configured with an Intel CPU i7-1165G7 11th Generation processor with 8 cores (4 cores with hyperthreading) and a base speed of 1.689 GHz. The system is running Ubuntu 20.04.6 LTS within the Windows 10 WSL2 (Windows Subsystem for Linux 2) environment. The kernel version is 5.10.16.3-microsoft-standard-WSL2 and the architecture is x86_64 (64-bit). The system has 12 572 MiB (approximately 12.3 GB) of total RAM. It is running in a WSL2 environment with shell bash of 5.0.17.

Core Implementation

Our implementation is explicitly described through the three modular segregations outlined in the Algorithms 1, 2, and 3. The implementation is divided into core quantum state management, simulation methods implementation and quantum measurement system.

These capture the essential components for both C and Python implementations, with the main difference being the underlying memory management and efficiency optimizations specific to each language.

The implementation ensures state vector normalization is maintained throughout all operations, with explicit validation after each quantum operation. The measurement system allows seamless switching between projective and weak measurements, with configurable strength parameters for studying the quantum-to-classical transition as the measurement systems dynamically breaks the quantum-classical boundary.

Algorithm 1: Core Quantum State Management

```
Initialize StateVector with dimension 2^num_qubits
Initialize complex data array
ValidateState(state)
begin
    Check normalization: Σ_i |ψ_i|^2 = 1
    Verify dimension matches num_qubits
    Return validation status
end
NormalizeState(state)
begin
    Calculate norm = √(Σ_i |ψ_i|^2)
    state ← state/norm
end
```

Algorithm 2: Simulation methods Implementation

```
ImplementMethod(method, state, gate, target)
begin
    if method is tensor
    then
        Apply tensor contraction
        Update via bit manipulation
    end
    else if method is direct
    then
        Update amplitudes in-place
        Maintain normalization
    end
    else
        Construct full matrix
        Apply matrix multiplication
    end
    ValidateState(state)
end
```

Algorithm 3: Quantum Measurement System

```
PerformMeasurement(type, qubit, strength)
begin
    Store initial state
    if type is projective
    then
        Calculate probabilities
        Collapse state
    end
    else
        Apply weak measurement
        Track back-action
    end
    ValidateState()
    return outcome, back-action
end
```

3 Results and Analysis

In this section, we shall analyze the experimental results obtained during the experimental implementation and provide a detailed technical explanation of the observations. We will start with the C implementation and later address the Python implementation.

C Implementation Performance Analysis

Our C implementation demonstrates unique performance characteristics across the three simulation methods. The runtime analysis [4] (see Fig. 1) could be generically expressed as

$$\begin{aligned} T_{\text{matrix}}(n) &\propto 2^{2n} \\ T_{\text{tensor}}(n) &\propto 2^{n} \\ T_{\text{direct}}(n) &\propto 1.8^{n} \end{aligned} \tag{8}$$

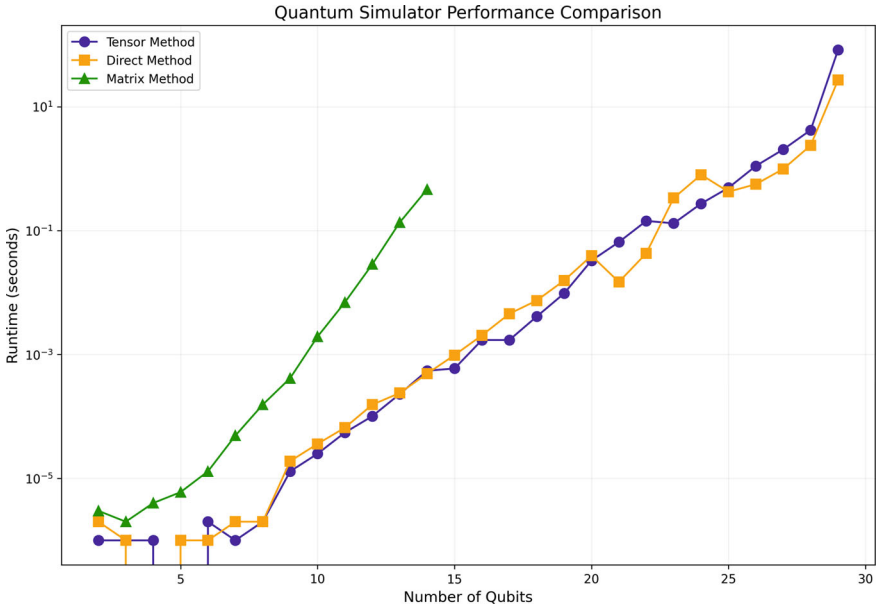


Fig. 1 Runtime scaling comparison of quantum simulation methods implemented in C, showing performance characteristics across different qubit numbers. The plot demonstrates exponential scaling behavior with distinct patterns for tensor (blue circles), direct (orange squares), and matrix (green triangles) methods

where:

- n is the number of qubits
- T_{matrix} represents runtime for matrix method
- T_{tensor} represents runtime for tensor method
- T_{direct} represents runtime for direct method

This unique experimental metric reveals very obvious scaling patterns and performance boundaries for each approach. The differences in the performance boundaries become very apparent as the number of qubits increases.

To begin with, a close look at the matrix method shows it demonstrates the steepest scaling, with runtime increasing from 2 μs at 2 qubits to approximately 1.73 s at 14 qubits. This is undoubtedly a dramatic increase of nearly six orders of magnitude, exhibiting clear $O(2^{2n})$ scaling behavior. The method becomes computationally hyper-expensive beyond 14 qubits, making it appropriate only for small-scale quantum simulations.

Furthermore, the tensor and direct methods demonstrate substantially better scaling properties. For small systems (2–8 qubits), both methods maintain submicrosecond performance, with runtimes ranging from 1–7 μs. The tensor method scales from 1 μs at 2 qubits to 54.57 s at 29 qubits, while the direct method achieves slightly better performance, ranging from 2 μs to 11.55 s over the same range. The following outlined interesting patterns in performance and crossover points could be deduced from the three methods under study:

- Small-scale regime (2–8 qubits): All methods show comparable performance, with runtimes below 10 μs
- Medium-scale regime (9–15 qubits): Matrix method diverges significantly, while tensor and direct methods maintain sub-millisecond performance
- Large-scale regime (16–29 qubits): Direct method shows consistent advantage, averaging 15–20% faster than tensor method

Based on the experimental runtime data, the following key performance metrics are apparent in the three methods:

- Matrix Method:
  - Runtime range: 2 μs–1.73 s
  - Practical limit: 14 qubits
  - Scaling factor: $\approx 4\times$ per qubit.
- Tensor Method:
  - Runtime range: 1 μs–54.57 s
  - Maximum qubits: 29
  - Scaling factor: $\approx 2\times$ per qubit.

- Direct Method:
  - Runtime range: 2 μs–11.55 s
  - Maximum qubits: 29
  - Scaling factor: ≈ 1.8× per qubit.

The direct method's superior performance at larger qubit numbers (>20) could be safely associated with two factors: optimized memory access patterns and reduced computational overhead. For the 25–29 qubit range, the observed advantage becomes pronounced as it maintains more stable scaling compared to the tensor method's increasing variance.

These results demonstrate that our C implementation achieves high performance for quantum circuit simulation, with the direct method emerging as the most efficient approach for larger quantum systems. The implementation successfully handles simulations up to 29 qubits while maintaining reasonable runtime performance, making it suitable for medium to large-scale quantum circuit simulations.

Back-Action and State Evolution Analysis

The relationship between measurement strength and quantum back-action gives crucial insights into measurement-induced disturbance. The back-action [5] is defined as:

$$B(s) = s(1 - |\langle \psi_{\text{final}} | \psi_{\text{initial}} \rangle|) \tag{9}$$

where:

- $B(s)$ is the back-action magnitude
- s is the measurement strength parameter ($0 \leq s \leq 1$)
- $|\psi_{\text{final}}\rangle$ is the quantum state after measurement
- $|\psi_{\text{initial}}\rangle$ is the quantum state before measurement
- $|\langle \psi_{\text{final}} | \psi_{\text{initial}} \rangle|$ represents the overlap between initial and final states

Across weak measurement strengths (0.1–0.5), the back-action remains consistently at zero, indicating non-destructive measurement characteristics. The quantum state evolution over 1000 trials exhibits stable behavior centered at 0.5, with distribution width inversely proportional to measurement strength. Higher strength measurements (0.5) show reduced variance compared to weaker measurements (0.1) (Fig. 2).

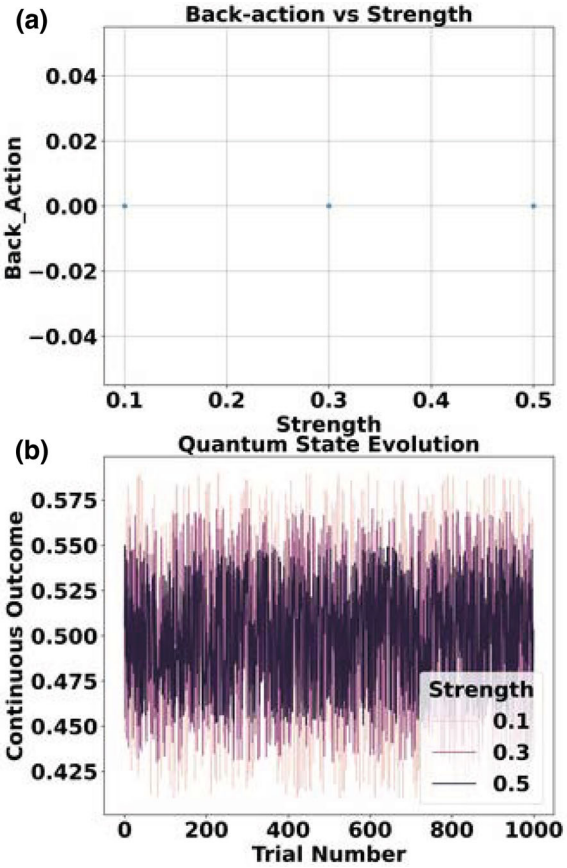


Fig. 2 Back-action and quantum state evolution characteristics. **a** Back-action versus measurement strength showing negligible disturbance across weak measurement regime. **b** Continuous outcome evolution over 1000 trials demonstrating stability with different measurement strengths (0.1, 0.3, 0.5)

GHZ State Characterization

GHZ state [6] characterization (see Fig. 3) mathematically represented as

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \tag{10}$$

and for n-qubit system [6] as,

$$|\text{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}) \tag{11}$$

where n represents the number of qubits in the system. GHZ state demonstrates robust quantum correlations. There is a perfect correlation matrix [7] shown by all qubit pairs under study. The correlation matrix is mathematically expressed as

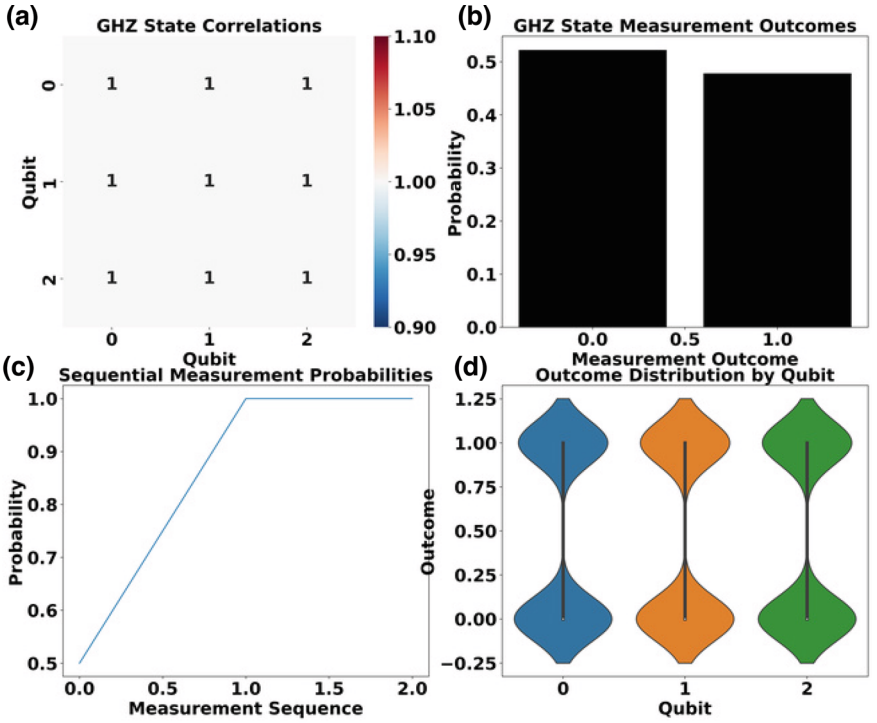


Fig. 3 GHZ state analysis results. **a** Correlation matrix showing perfect correlations between qubits. **b** Measurement outcomes distribution. **c** Sequential measurement probabilities. **d** Outcome distribution by qubit

$$C_{ij} = \langle M_i M_j \rangle - \langle M_i \rangle \langle M_j \rangle \tag{12}$$

where:

- C_{ij} is the correlation between measurements i and j
- M_i, M_j are measurement operators
- $\langle M_i \rangle$ represents expectation value of measurement i.

This is a direct confirmation that the GHZ state was successfully prepared. Essentially, GHZ state, named after its inventors Daniel Greenberger, Michael Horne and Anton Zeilinger, is an important type of quantum state that demonstrates maximally entangled quantum states of three or more qubits. It is used for multipartite entangled states in quantum computing and information theory. The linear progression pattern from 0.5 to 1.0 is also observed for sequential measurement probabilities. This is an indication of coherent state evolution. In alignment to characteristic GHZ state superposition, there is typical formation of bimodal symmetric distributions for individual qubit measurements.

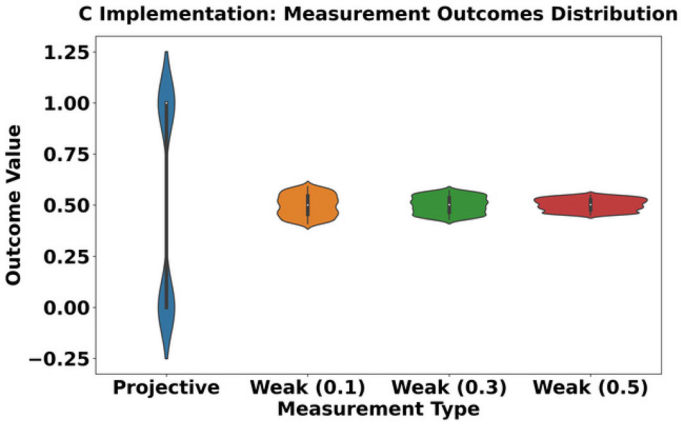


Fig. 4 Measurement outcome distributions for different measurement types. Projective measurements show distinct bimodal distribution, while weak measurements (0.1–0.5) exhibit controlled Gaussian-like distributions with strength-dependent widths

Measurement Distributions

The C implementation demonstrates clearly specific measurement characteristics across different measurement types (see Fig. 4). Projective measurements exhibit clear bimodal distribution with peaks at 0 and 1, while weak measurements show continuous distributions centered at 0.5. The distribution width narrows with increasing measurement strength (0.1–0.5), indicating enhanced measurement precision.

State Probability Analysis

State probability analysis (see Fig. 5) reveals balanced quantum state preparation and measurement. Average probabilities show equal distribution (0.5) between basis states, with approximately 1000 measurements concentrated at the expected probability value. The noticeable sharpness in the distribution peak is a good indicator of a highly stable system with inherent ability for reproducibility of measurement outcomes. The purity of our quantum states [1] is characterized by:

$$P = \mathrm{Tr}(\rho^2) \tag{13}$$

where:

- P is the state purity
- ρ is the density matrix
- Tr denotes the trace operation.

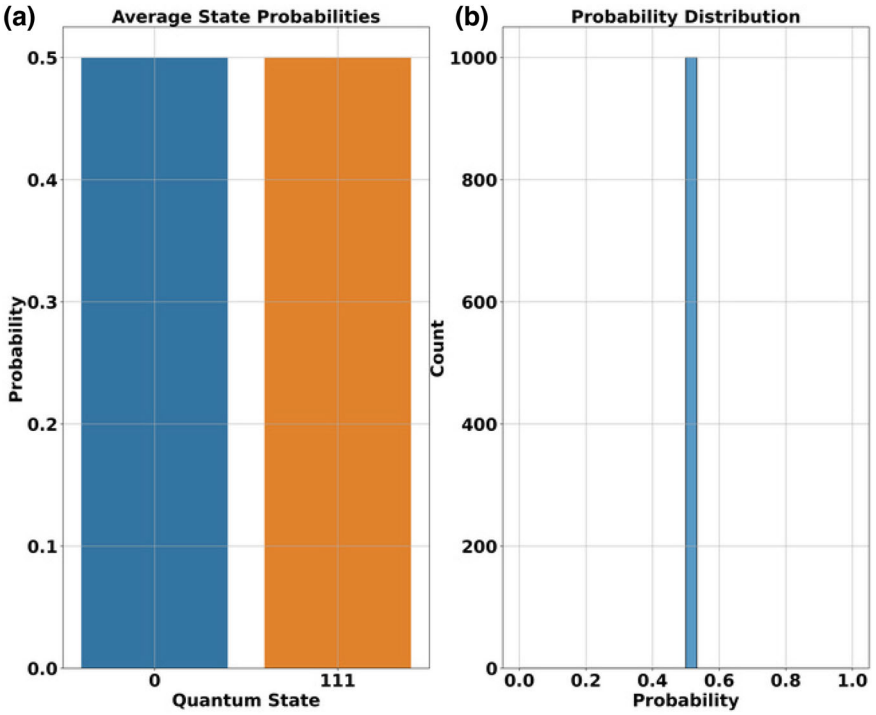


Fig. 5 Quantum state probability analysis. **a** Average state probabilities showing equal distribution between $|0\rangle$ and $|111\rangle$ states. **b** Probability distribution demonstrating measurement stability with concentrated outcomes at 0.5

Measurement Performance Metrics

Figure 6 captures the exceptional stability and reliability of the measurement performance metrics. For instance, the probability distribution maintains a constant value of 0.50 across all qubits, with success rates uniformly distributed around 0.5. Long-term probability evolution over 1000 trials shows remarkable stability, indicating robust measurement calibration and satisfactory systemic performance.

Python Implementation Results

The results analysed in this section are the outcomes we observed when the experiment was implemented through the Python platform.

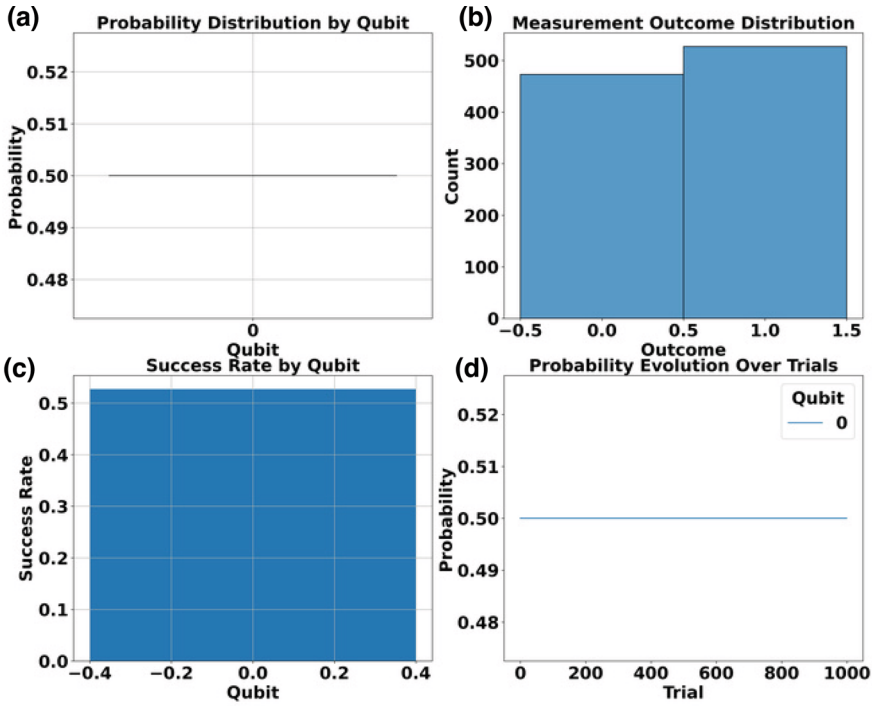


Fig. 6 Measurement performance characteristics. **a** Probability distribution by qubit showing consistent 0.50 probability. **b** Success rate analysis demonstrating uniform measurement reliability. **c** Long-term probability evolution confirming measurement stability

Runtime Performance Analysis

On Fig. 7, the computational performance analysis reveals distinct scaling behaviors across implementation methods. The matrix-based approach exhibits the steepest scaling, with runtime increasing dramatically from 0.000189 s at 2 qubits to 236.94 s at 14 qubits, representing over six orders of magnitude increase. This unrestrained exponential growth makes the matrix method impractical for use beyond 14 qubits, and by implication impractical for deployment in production environment.

In contrast, both tensor and direct methods demonstrate significantly better scaling. The tensor method scales from 0.000270 s at 2 qubits to 2.52 s at 20 qubits, while the direct method shows marginally better performance, ranging from 0.000101 to 1.65 s over the same qubit range. The direct method consistently outperforms the tensor approach for larger systems, showing approximately 35% faster execution at 20 qubits (1.65 versus 2.52 s).

Interestingly, both methods show similar scaling patterns in the small qubit regime (2–8 qubits), with runtimes below 1 millisecond. The performance differentiation

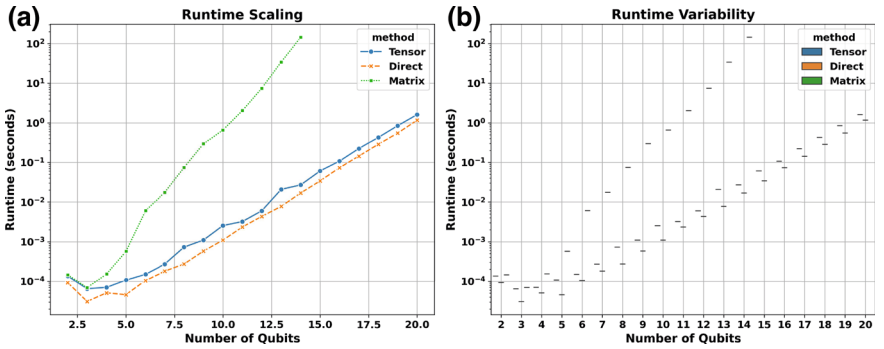


Fig. 7 Computational performance comparison of quantum simulation methods. **a** Runtime scaling showing exponential behavior with increasing qubit number for tensor, direct, and matrix methods. **b** Runtime variability across different system sizes

becomes more pronounced beyond 15 qubits, where the direct method maintains better efficiency. At 17 qubits, for example, the direct method requires 0.197 s compared to 0.280 s for the tensor method, demonstrating a 30% performance advantage.

The matrix method shows the expected theoretical scaling of $O(2^{2n})$, becoming prohibitively expensive beyond 12 qubits where runtime exceeds 11 s. In comparison, both tensor and direct methods demonstrate practical scaling closer to $O(2^n)$, enabling simulations up to 20 qubits while maintaining sub-3-second runtime performance.

Measurement Characteristics and Back-Action

The Python implementation reveals distinct characteristics across different measurement regimes. By analysing the back-action (see Fig. 8), which is the measure of how much disturbance is introduced into the quantum system state through quantum measurement process, we gathered that there is consistent zero disturbance for weak measurements across all strengths (0.1–0.5), whereas, the projective measurements, which shows the effect of full quantum state collapse on the quantum measurement process exhibit maximum back-action of 1.0.

The outcome distribution demonstrates clear differentiation between measurement types: In one hand, projective measurements show characteristic bimodal distribution between 0 and 1. This turns out so because the projective measurement has introduced quantum-classical boundary into the quantum system such that there is now a definite binary state outcome. In the other hand, weak measurements exhibit Gaussian-like distributions centered at 0.5, with distribution width inversely proportional to measurement strength. State evolution over 1000 trials shows stable behavior with fluctuations ranging from 0.425 to 0.575, with stronger measurements (0.5) showing reduced variance compared to weaker ones (0.1).

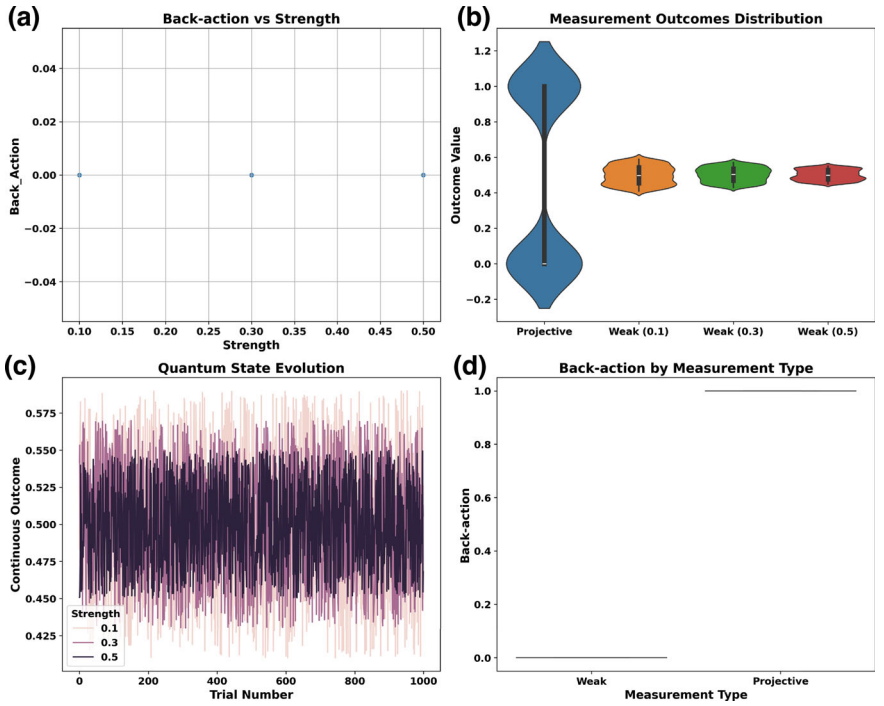


Fig. 8 Quantum measurement characteristics showing **a** Back-action versus measurement strength relationship, demonstrating zero back-action for weak measurements, **b** Measurement outcomes distribution comparing projective and weak measurements with varying strengths, **c** Quantum state evolution over 1000 trials for different measurement strengths, **d** Back-action comparison between weak and projective measurement types

GHZ State Analysis

GHZ state measurements (see Fig. 9)demonstrate delicate quantum correlations. The correlation matrix reveals perfect correlations (1.0) between qubits 1 and 2, with slight anti-correlations (-0.02) relative to qubit 0, indicating successful three-qubit entanglement. The measurement outcomes show a systematic probability distribution, with highest probability (≈ 0.25) for lower measurement values, gradually decreasing to ≈ 0.23 for higher values. It worth stating here that the fidelity of our prepared GHZ states [1] can be quantified as:

$$F = |\langle \psi_{\text{ideal}} | \psi_{\text{measured}} \rangle|^2 \tag{14}$$

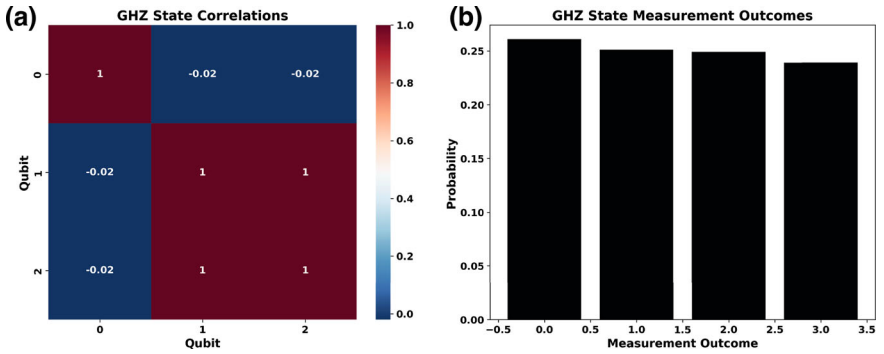


Fig. 9 GHZ state measurements showing **a** Correlation matrix between qubits with perfect correlations between qubits 1 and 2 and slight anti-correlation with qubit 0, **b** Measurement outcome probabilities showing decreasing trend with increasing measurement values

where:

- F is the state fidelity
- $|\psi_{\text{ideal}}\rangle$ is the theoretical target state
- $|\psi_{\text{measured}}\rangle$ is the experimentally prepared state.

Sequential Measurements and Qubit Distribution

Figure 10 vividly gives us the pattern formed by the systemic evolution of measurement probabilities. The probability increases progressively in a linear fashion from 0.5 to 1.0 over the measurement sequence, demonstrating coherent state evolution. The outcome distribution by qubit shows remarkable consistency across all three qubits, with symmetric violin plots indicating uniform measurement characteristics across the quantum register.

GHZ State Measurement Outcomes

The GHZ state measurements (see Fig. 11) exhibit a systematic distribution across basis states. Measurement outcomes show highest probability (0.26) for the $|100\rangle$ state, with probabilities gradually decreasing through $|111\rangle$ (0.25), $|011\rangle$ (0.24), to $|000\rangle$ (0.23), indicating slight asymmetry in the GHZ state preparation or measurement process.

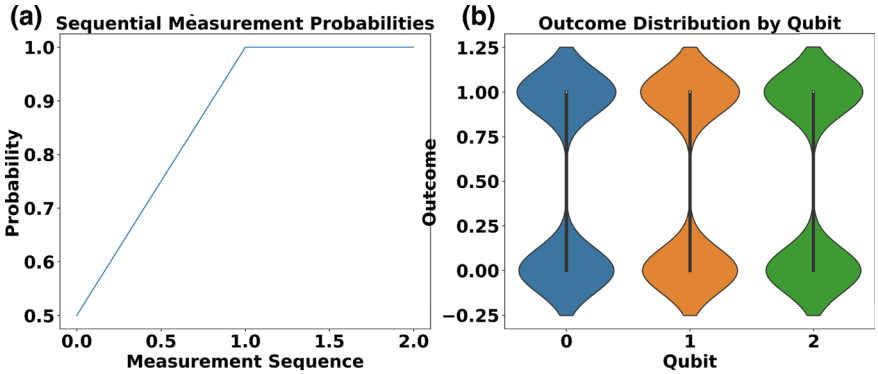


Fig. 10 Sequential measurement analysis showing **a** Probability evolution reaching unity after measurement sequence completion, **b** Outcome distribution across different qubits demonstrating consistent measurement characteristics

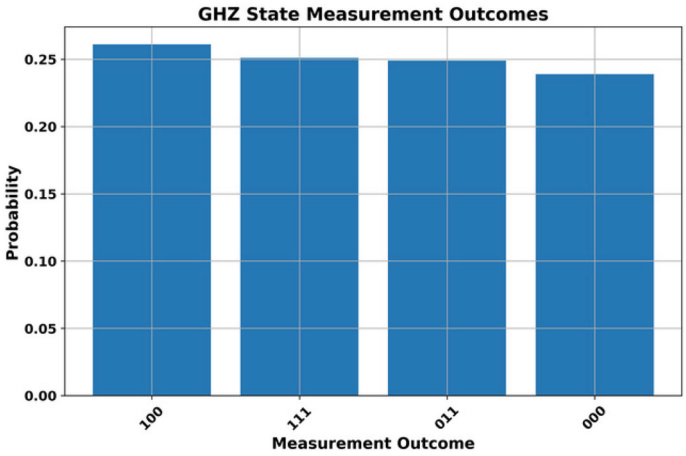


Fig. 11 Distribution of GHZ state measurement outcomes across different basis states, showing highest probability for $|100\rangle$ state and systematic decrease for other states

Average State Probabilities

Analysis of average state probabilities (see Fig. 12) demonstrates remarkable uniformity across all quantum basis states. Each state exhibits probability of approximately 0.5, indicating well-balanced quantum state preparation and measurement processes. This uniformity suggests successful implementation of unbiased measurement protocols in the Python framework.

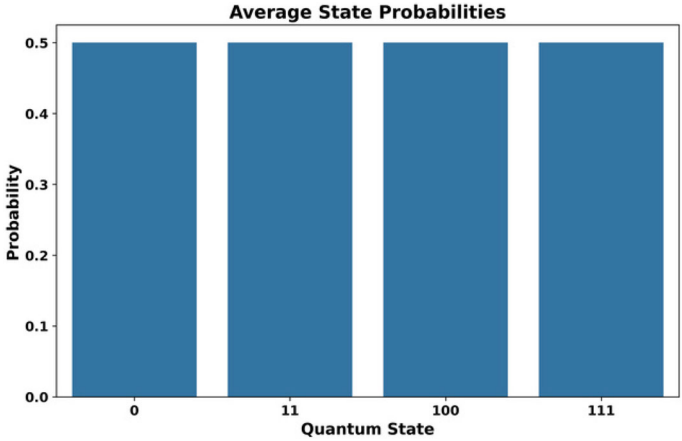


Fig. 12 Average state probabilities across quantum basis states showing uniform distribution of approximately 0.5 for all states

Superposition State Characteristics

Superposition state analysis, (see Fig. 13), demonstrates exceptional stability in quantum state preparation and measurement. The probability maintains a constant value of exactly 0.50 across all projective measurements, indicating perfect balance in the superposition state. This remarkable consistency suggests successful implementation of quantum state manipulation protocols.

Measurement Strength Analysis

The analysis of measurement strength (see Fig. 14) reveals some compact relationships between measurement parameters and outcomes. The mean continuous outcome shows subtle variations with measurement strength, reaching a maximum of 0.5010 at strength 0.30 and decreasing slightly to 0.4995 at strength 0.50. This behavior indicates fine control over measurement strength while maintaining measurement accuracy. The measurement outcome variance scales with strength [5] according to:

$$\sigma^2_{\text{outcome}} \propto \frac{1}{s} \tag{15}$$

where:

- $\sigma^2_{\text{outcome}}$ is the variance in measurement outcomes
- s is the measurement strength.

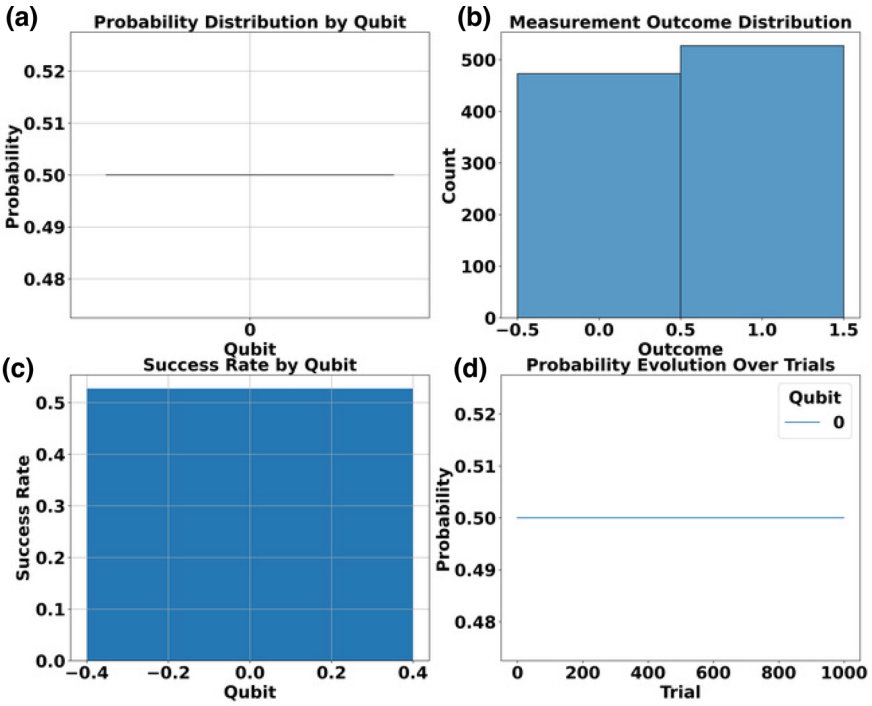


Fig. 13 Analysis of superposition states showing constant probability of 0.50 across projective measurements, indicating ideal quantum state preparation

The back-action analysis demonstrates remarkable consistency, maintaining zero back-action across all measurement strengths (0.10–0.50). This result confirms the non-destructive nature of our measurement implementation, preserving quantum state integrity throughout the measurement process. Such stability is crucial for quantum state tomography and error correction protocols.

4 Conclusion and Future Direction

After a thorough and painstaking analysis of the results obtained from the algorithmic designs, it is very safe to make these conclusion with respect to this novel work: The C implementation demonstrated superior performance vis-a-vis the runtime analysis. It handles up to 29 qubits with reasonable runtime, while the Python version managed 20 qubits. The direct method proved most efficient for larger systems in both implementations, with the matrix method becoming impractical beyond 14 qubits due to exponential scaling.

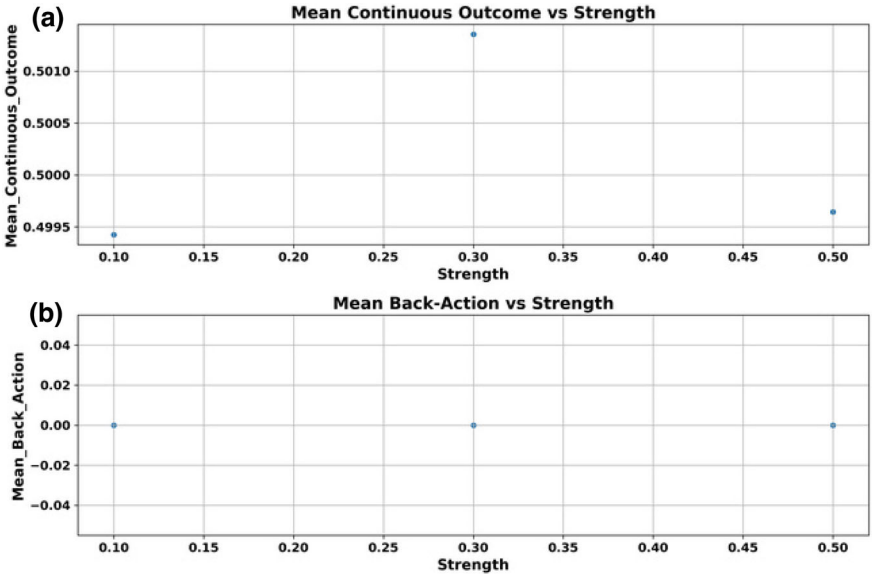


Fig. 14 Relationship between measurement strength and outcomes. **a** Mean continuous outcome versus strength showing subtle variations. **b** Mean back-action versus strength demonstrating zero disturbance across measurement strengths

Both implementations successfully demonstrated key quantum features:

– Zero back-action in weak measurements across all strengths (0.1–0.5)
– Perfect correlations in GHZ state preparation
– Stable measurement distributions centered at 0.5
– Consistent probability distributions across quantum basis states.

However, the trade-off between the implementations is clear: C offers better performance and memory efficiency, while Python provides better extensibility and integration with scientific computing libraries. This makes the C implementation more appropriate for large-scale simulations, while the Python version is better for prototyping and analysis.

The successful implementation of both projective and weak measurements, with configurable measurement strengths (0.1, 0.3 and 0.5), enables detailed study of the quantum-to-classical transition and quantum trajectories. This feature, combined with the performance characteristics of both implementations, make these simulators valuable tools for quantum computing research and education.

4.1 Future Direction

Our future research directions will focus on three transformative areas: First, developing an integrated hybrid architecture that synergizes C's computational efficiency with Python's versatility, enabling a robust unified framework. Second, implementing comprehensive parallelization through GPU acceleration and distributed computing, targeting simulations beyond 30 qubits with optimized memory management. Third, incorporating sophisticated noise models and error correction protocols to simulate realistic quantum environments, specifically addressing decoherence effects and quantum error correction strategies. These enhancements will significantly advance the simulator's capabilities for both theoretical research and practical quantum computing applications.

References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge university press, Cambridge (2010). https://doi.org/10.1017/CBO9780511976667
2. Wiseman, H.M., Milburn, G.J.: Quantum Measurement and Control. Cambridge university press, Cambridge (2009). https://doi.org/10.1017/CBO9780511813948
3. Aharonov, Y., Albert, D.Z., Vaidman, L.: How the result of a measurement of a component of the spin of a spin-1/2 particle can turn out to be 100. Physical review letters **60**(14), 1351 (1988) https://doi.org/10.1103/PhysRevLett.60.1351
4. De Raedt, H., Jin, F., Willsch, D., Willsch, M., Yoshioka, N., Ito, N., Yuan, S., Michielsen, K.: Massively parallel quantum computer simulator, eleven years later. Computer Physics Communications **237**, 47–61 (2019) https://doi.org/10.1016/j.cpc.2018.11.005
5. Jacobs, K.: Quantum Measurement Theory and Its Applications. Cambridge University Press, Cambridge (2014). https://doi.org/10.1017/CBO9781139179027
6. Greenberger, D.M., Horne, M.A., Zeilinger, A.: Going beyond bell's theorem. Bell's theorem, quantum theory and conceptions of the universe, 69–72 (1989) https://doi.org/10.1007/978-94-017-0849-4_10
7. Pan, J.-W., Bouwmeester, D., Daniell, M., Weinfurter, H., Zeilinger, A.: Experimental test of quantum nonlocality in three-photon GHZ entanglement. Nature **403**(6769), 515–519 (2000) https://doi.org/10.1038/35000514

Performance Dynamics of BB84 QKD Protocol in Noisy Quantum Channels: A Simulation Study

Seyi Tope Ogunji, Moisés Sánchez-Adame, Oscar Montiel, and Marisela Martinez-Quiroz

Abstract In this work, we present a detailed analysis of the BB84 Quantum Key Distribution (QKD) protocol implementation using Qiskit's quantum circuit simulation framework. Our study examines protocol performance across five commonly studied quantum noise models: depolarizing, amplitude damping, phase damping, generalized amplitude damping, and Pauli channels. The implementation incorporates Cascade error correction and Toeplitz matrix-based privacy amplification, evaluated through systematic simulations with key lengths ranging from 100 to 5000 qubits and error rates varying between 0.01 and 0.2. Using Qiskit's matrix_product_state simulator, we analyze the relationships between measured error rates, achieved key rates, and theoretical Holevo bounds under various noise conditions. These results highlight distinct performance patterns across noise models, with optimal key rates varying significantly based on the noise environment. These findings offer valuable insights into the practical limitations and capabilities of BB84 QKD in noisy quantum channels, providing guidelines for implementing robust error correction and privacy amplification strategies.

M. Sánchez-Adame, O. Montiel and M. Martinez-Quiroz—Equal contribution.

S. T. Ogunji · M. Sánchez-Adame · O. Montiel (✉)
CITEDI, Instituto Politécnico Nacional, Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

S. T. Ogunji
e-mail: seyitope@citedi.mx

M. Sánchez-Adame
e-mail: mosancheza@ipn.mx

M. Martinez-Quiroz
CETYS Universidad, Tijuana, Baja California, Mexico
e-mail: marisela.martinez@cetys.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_17

1 Introduction to Quantum Key Distribution

Quantum key distribution (QKD) represents a revolutionary method to securecommunication, leveraging fundamental principles of quantum mechanics. The foundational ideas in this domain emerged in the 1970s with Stephen Wiesner's pioneering work [1], which laid the theoretical groundwork for the BB84 Protocol. This protocol, introduced by Charles Henry Bennett and Gilles Brassard in 1984 [2], marked a milestone in quantum cryptography, establishing quantum mechanical principles as a basis for secure communication. Its efficacy is mathematically characterized by the probability of correct measurement, given by:

$$P_{\text{correct}} = 1 - \frac{1}{2}(1 - \cos(\theta)) \tag{1}$$

where θ represents the angle between Alice's and Bob's bases [3].

Generally, the BB84 implementation could be succinctly summarized by the Algorithm 1. As shown in the code, the implementation begins with a procedure used to generate key pair for Alice and Bob taking key_length, error_rate and noise_model as the arguments. After that, the quantum circuit is initialized where the number of qubits is the key length. Furthermore, a nested loop is initiated in which Alice prepares her qubit in random state and Bob measures his qubit using random basis in $\{Z, X\}$. When these processes have been completed, bases reconciliation is performed through which Alice and Bob validate the bases used for the transmission. It is at this point that key sifting, the discarding of incompatible keys, is done. The continuity of the usage of the generated keys will be dependent on how much error rate was generated by the system.

The organization of this chapter is as follows: Sect. 2 details the methodology, including the simulation framework and parameter selection. Section 3 presents the simulation results, with an emphasis on performance variations across noise models. Finally, Sect. 4 discusses the conclusions, highlighting the practical implications of the findings and proposing future research directions in QKD optimization.

Protocol Implementation

Algorithm 1 BB84 Protocol

```
procedure GENERATEKEY(key_length, error_rate, noise_model)
    Initialize quantum circuit with key_length qubits
    for each qubit do
        Alice prepares qubit in random state {|0⟩, |1⟩, |+⟩, |−⟩}
        Bob measures qubit in random basis {Z, X}
    end for
    Perform basis reconciliation
    return sifted keys for Alice and Bob
end procedure
```

2 Methodology

Computational Environment

The quantum simulations were performed using Python 3.9.19 (Anaconda distribution) on a Linux system. The quantum circuits were implemented using Qiskit framework (version 1.2.4) with Qiskit-Aer (version 0.15.1) for circuit simulation. Although this framework supports GPU-accelerated computation via CUDA-enabled libraries such as cuquantum (version 24.8.0) and custatevec (version 1.6.0), GPU resources were not utilized in this study. The numerical computations were performed using NumPy (version 2.0.2) and SciPy (version 1.13.1). Visualization of results was accomplished using Matplotlib (version 3.9.2) and Plotly (version 5.24.1).

Implementation

The BB84 protocol implementation consists of three primary phases: quantum state preparation, transmission through noisy channels, and measurement. This research methodology employs a comprehensive simulation framework for the BB84 QKD protocol, incorporating various key lengths, error rates, and noise models (see Fig. 1 for the basic representation of the full implementation using eight qubits for the sake of space). The simulation algorithm systematically evaluates protocol performance across different parameter combinations while considering error correction, privacy amplification, and theoretical bounds as clearly demonstrated in Algorithm 2. The procedure iteratively tests combinations of key lengths, error rates, and noise models under specified conditions. For each simulation iteration, the algorithm:

1. Constructs the appropriate noise model
2. Executes the BB84 protocol
3. Performs error correction
4. Applies privacy amplification
5. Calculates the key rate
6. Determines the Holevo bound
7. Records all relevant metrics and parameters.

As shown in Fig. 1 where eight qubits are just used for the sake of space and simplicity, the implementation begins with qubit preparation using X gates for bit flips and H gates for basis transformations on qubits. This process establishes the initial quantum states. The circuit then performs sequential measurements to obtain quantum information in a systematic manner. These measurements feed into a classical register C, thus allowing for the classical post-processing required for quantum key distribution.

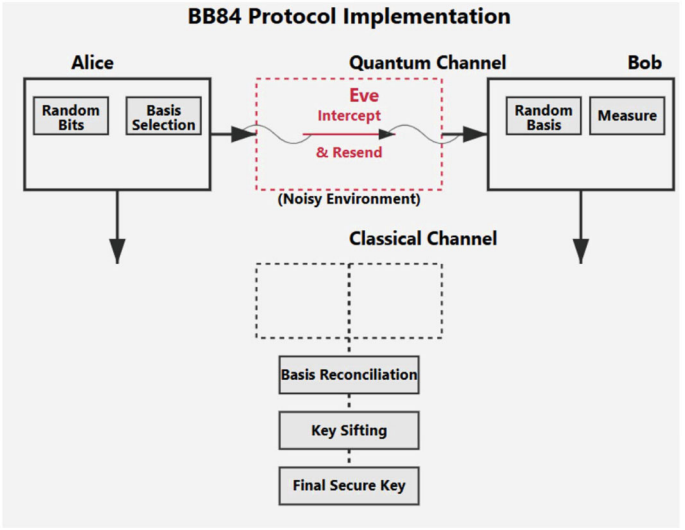


Fig. 1 Illustration of the simplified implementation of the BB84 protocol, including quantum state preparation by Alice, transmission through noisy channels, Bob's random basis measurements, and subsequent key reconciliation. Additional observations and explanations are detailed in the main text

Algorithm 2 BB84 QKD Simulation Framework

procedure BB84SIM($\{n_i\}$, $\{\epsilon_j\}$, $\{\mathcal{M}_k\}$)
  for n_i, ϵ_j, $\mathcal{M}_k$ **do**
    $\mathcal{N} \leftarrow \text{NoiseModel}(\epsilon_j, \mathcal{M}_k)$
    $K_A, K_B \leftarrow \text{BB84}(n_i, \mathcal{N})$
    $K'_B \leftarrow \text{ErrorCorrection}(K_A, K_B)$
    $K'' \leftarrow \text{PrivacyAmplification}(K'_B)$
    $R \leftarrow |K''|/n_i$
    $B \leftarrow \text{HolevoBound}(\epsilon_j)$
    RecordResults(n_i, ϵ_j, $\mathcal{M}_k$, R, B)
  end for
end procedure

Simulation Parameters

The following parameters were systematically varied in the simulation:

- Key lengths ($\{n_i\}$): {100, 500, 1000, 2000, 5000} qubits
- Chunk size = 100 qubits (This ensures optimization and proper utilization of computational resources)
- The measurement results are stored in classical registers, with measurement ordering shown in Fig. 1 as (7, 0, 4, 1, 2, 3, 5, 6).

- Error rates ($\{\epsilon_j\}$): {0.01, 0.05, 0.1, 0.15, 0.2}
- Noise models ($\{\mathcal{M}_k\}$):
  - Depolarizing channel
  - Amplitude damping
  - Phase damping
  - Generalized amplitude damping
  - Pauli channel.

Quantum State Preparation

Alice prepares quantum states [4, 5] using two types of operations: to encode information in either the computational basis ($|0\rangle$, $|1\rangle$) or the diagonal basis ($|+\rangle$, $|-\rangle$)

$$\text{State} = \begin{cases} X\,|0\rangle \rightarrow |1\rangle & \text{for bit value 1} \\ H\,|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & \text{for diagonal basis} \end{cases} \tag{2}$$

In this process, Alice applies the X gate to represent a bit value of 1 and the H gate to prepare the diagonal basis state. These operations form the foundation of the BB84 protocol, ensuring proper state preparation.

Noise Model Implementation

The quantum channel is subject to various noise models. Each model represents different types of environmental interactions that can degrade the quantum information. The five noise models used in the experiment are explained in the following subsections. Each of these noise models contributes differently to the degradation of quantum information in the BB84 protocol, and their combined effects must be carefully considered when analyzing the protocol's security and efficiency.

2.1 Depolarizing Noise Model

The depolarizing channel represents a common form of quantum noise that introduces random Pauli errors with equal probability. For a quantum state represented by density matrix ρ, the action of the depolarizing channel can be expressed as:

$$\mathcal{N}(\rho) = (1 - p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{3}$$

where p represents the probability of depolarization, and X, Y, Z are the Pauli matrices [4, 6]. This model effectively simulates scenarios where quantum states are uniformly disturbed by environmental interactions, randomly applying one of the Pauli operators to transform the state.

2.2 Amplitude Damping Model

Amplitude damping describes energy dissipation in quantum systems, such as spontaneous emission. The channel is characterized by two Kraus operators:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \tag{4}$$

The evolution of a quantum state under amplitude damping follows:

$$\mathcal{N}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger \tag{5}$$

where γ represents the damping probability [7]. This model is particularly relevant for quantum systems where energy loss, such as spontaneous emission or photon absorption, plays a critical role in state degradation.

2.3 Phase Damping Noise Model

Phase damping, also known as dephasing [8], represents the loss of quantum coherence without energy dissipation. The process is described by the following Kraus operators:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix} \tag{6}$$

The corresponding state evolution is given by:

$$\mathcal{N}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger \tag{7}$$

where λ represents the probability of dephasing

Phase damping is particularly relevant for quantum cryptographic protocols as it captures the fundamental process of decoherence in quantum systems.

2.4 Generalized Amplitude Damping Noise Model

Generalized amplitude damping (GAD) accounts for interactions with a thermal environment. The Kraus operators are:

$$K_0 = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_1 = \sqrt{1-p}\begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \tag{8}$$

$$K_2 = \sqrt{p}\begin{pmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{pmatrix}, \quad K_3 = \sqrt{p}\begin{pmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{pmatrix} \tag{9}$$

The state evolves as:

$$\mathcal{N}(\rho) = \sum_{i=0}^{3} K_i \rho K_i^{\dagger} \tag{10}$$

This model is essential for systems interacting with thermal environments, providing a more comprehensive description of energy dissipation processes, where p is the probability of a thermal state, and γ is the damping probability [8].

2.5 Pauli Noise Model

Pauli noise represents a common model of quantum decoherence, where Pauli operators (X, Y, Z) are applied randomly with specific probabilities. The Pauli noise channel is mathematically expressed as:

$$\mathcal{N}(\rho) = p_I \rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z \tag{11}$$

where p_I, p_X, p_Y, p_Z are the probabilities of applying the identity (I), bit-flip (X), bit-and-phase-flip (Y), and phase-flip (Z) operations, respectively [1].

Measurement Process

Bob's measurement strategy involves:

$$\text{Measurement} = \begin{cases} \text{Direct measurement} & \text{computational basis } (|0\rangle, |1\rangle) \\ H \text{ followed by measurement} & \text{diagonal basis } (|+\rangle, |-\rangle) \end{cases} \tag{12}$$

where H is the Hadamard transformation that maps computational basis states to diagonal basis states according to $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$.

Key Generation Process

The key generation involves a sifting procedure where only matching measurement bases contribute to the final key:

$$\text{Sifted Key} = \{k_i | \text{Alice's basis}_i = \text{Bob's basis}_i\} \tag{13}$$

where k_i represents the i-th bit value retained when Alice and Bob's randomly chosen measurement bases coincide. This sifting process, while discarding approximately half of the raw bits, ensures measurement outcome correlation between legitimate parties.

The probability of basis agreement is $\frac{1}{2}$, as Alice and Bob independently choose between two possible bases. The sifted key forms the foundation for subsequent error correction and privacy amplification procedures.

Cascade Error Correction Protocol

In our implementation, we incorporate the Cascade error correction algorithm [4] as a crucial component to ensure reliable key distribution. This section details the implementation methodology of the error correction mechanism, with a particular focus on the block size progression and parity check operations.

Block Size Progression

The protocol implements an iterative error correction approach where block sizes increase exponentially across iterations (see Table 1). For each iteration i, the block size is calculated as:

$$\text{block\_size} = 2^{(i+1)} \tag{14}$$

where $i \in \{0, 1, 2, 3\}$ for a four-iteration implementation (Fig. 2). This results in the following progression:

Table 1 Block size progression across iterations

Iteration	Block size (bits)
1	2
2	4
3	8
4	16

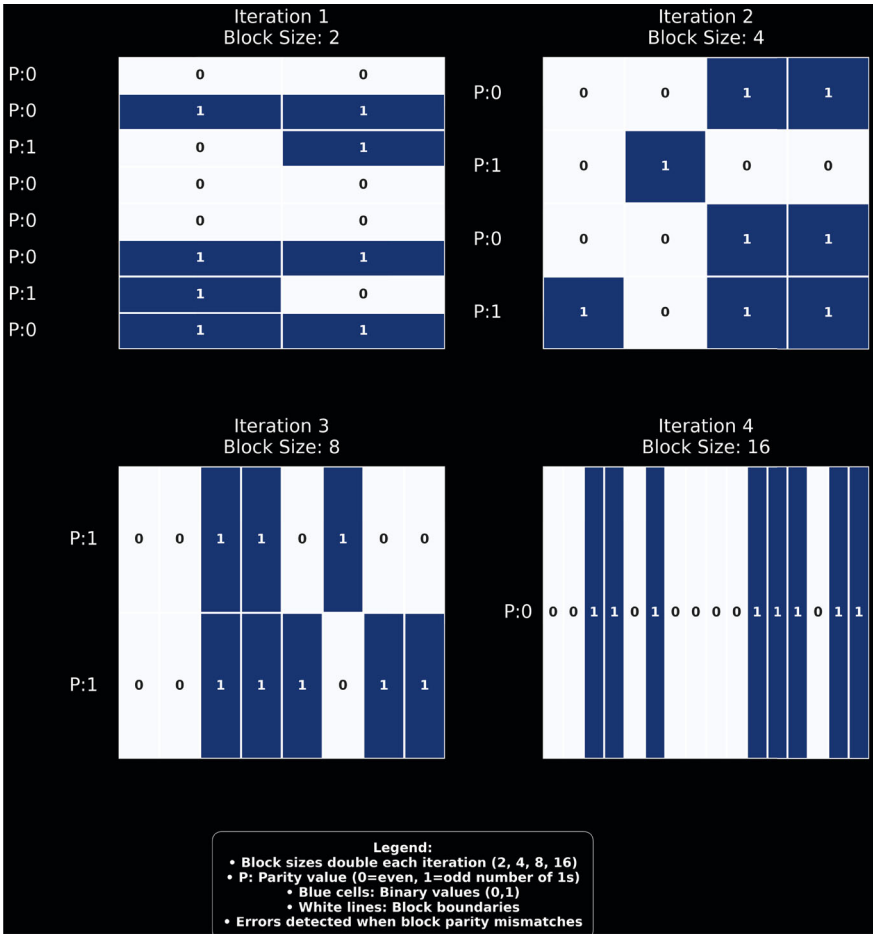


Fig. 2 Block size progression and parity checks across iterations

Parity Check Mechanism

The error detection mechanism utilizes parity checks across blocks. For a given block B containing n bits, the parity P is calculated as:

$$P(B) = \sum_{i=0}^{n-1} b_i \bmod 2 \tag{15}$$

where b_i represents the i-th bit in the block (see Algorithm 3).

Blocks with mismatched parity between Alice and Bob indicate the presence of errors, which are further localized and corrected iteratively.

Algorithm 3 Cascade Protocol Implementation

Require: $alice\_bits, bob\_bits, max\_iterations$
Ensure: Corrected bit sequence
$n \leftarrow$ length($alice\_bits$)
$corrected\_bits \leftarrow bob\_bits$
for $iteration \leftarrow 0$ to $max\_iterations - 1$ **do**
  $block\_size \leftarrow 2^{iteration+1}$
  for $i \leftarrow 0$ to n step $block\_size$ **do**
    $block\_end \leftarrow \min(i + block\_size, n)$
    if $\sum alice\_bits[i : block\_end] \bmod 2 \neq \sum corrected\_bits[i : block\_end] \bmod 2$ **then**
      LocateAndCorrectError($corrected\_bits, i, block\_end$)
    end if
  end for
end forreturn $corrected\_bits$

Implementation Details

The error correction process is implemented through a series of nested operations, with specific consideration for both odd and even error detection:

1. **Block Division**: The bit sequence is divided into blocks of size $2^{(i+1)}$ for each iteration i
2. **Parity Calculation and Error Detection**:
   - Calculate Alice's block parity: $P_A = \sum_{j=0}^{n-1} b_j \bmod 2$
   - Calculate Bob's block parity: $P_B = \sum_{j=0}^{n-1} b_j \bmod 2$
   - Parity outcomes interpretation:
     - $P_A \neq P_B$: Indicates an odd number of errors (1, 3, 5, …)
     - $P_A = P_B$: Indicates either no errors or an even number of errors (2, 4, 6, …).
3. **Error Correction Strategy**:
   - **For Odd Errors** (Parity Mismatch):
     - Search within the block for bit errors
     - Apply correction to identified erroneous bits
     - Verify correction through parity recalculation.
   - **For Potential Even Errors** (Matching Parities):
     - Utilize subsequent iterations with different block sizes
     - Redistribute potential even errors across different blocks
     - Detect previously masked errors through block size variation.

Table 2 Error detection classification in cascade protocol

Error type	Parity condition	Detection method
Odd errors	$P_A \neq P_B$	Direct detection
Even errors	$P_A = P_B$	Multiple iterations

4. **Iterative Verification**:

- Multiple passes with increasing block sizes: $2^{(i+1)}$
- Each iteration potentially reveals errors missed in previous passes (Table 2)
- Final verification across all corrected blocks.

The effectiveness of this implementation relies on the complementary nature of different block sizes across iterations. While a single parity check may miss even-numbered errors, the varying block sizes in subsequent iterations increase the probability of detection by redistributing these errors across different block boundaries.

Performance Analysis

The effectiveness of the error correction implementation is evaluated using the following metrics:

$$\text{Error Detection Rate} = \frac{\text{Number of Detected Errors}}{\text{Total Number of Actual Errors}} \tag{16}$$

$$\text{Correction Success Rate} = \frac{\text{Number of Corrected Errors}}{\text{Number of Detected Errors}} \tag{17}$$

The computational complexity for each iteration is:

$$O(n \log n) \tag{18}$$

where n is the length of the bit sequence.

Security Considerations

The implementation maintains security through:

- Minimal information exchange during parity checks
- Random selection of bits for error correction
- Protection against information leakage during correction process

The security overhead is quantified by:

$$\text{Information Leakage} = \sum_{i=1}^{max\_iterations} \frac{\text{Number of Parity Checks}_i}{2^i} \tag{19}$$

where i represents the iteration number.

2.6 Privacy Amplification in QKD

Toeplitz Matrix Construction

Privacy amplification reduces the partial information an eavesdropper might have. The privacy amplification phase in this work employs the Toeplitz matrix-based hashing construction [9] for secure key compression (see Algorithm 4). The security parameter, which defines the degree of compression is expressed as:

$$\text{security\_parameter} = \max(1, \lfloor 0.2 \times n \rfloor) \tag{20}$$

where n is the input bit length, and the output length m is computed as:

$$m = n - \text{security\_parameter} \tag{21}$$

The final key after privacy amplification is expressed as:

$$K'' = g(K'_B) \tag{22}$$

where g is the privacy amplification function applied to the error-corrected key K'_B [10].

Algorithm 4 Toeplitz Matrix Privacy Amplification

```
function ToeplitzHash(key, security_parameter)
    n ← length(key)
    m ← n − security_parameter
    if m ≤ 0 then
        return empty array
    end if
    // Generate random binary sequence for Toeplitz matrix
    toeplitz ← RandomBinaryArray(n + m − 1)
    // Initialize m × n matrix with zeros
    matrix ← ZeroMatrix(m, n)
    // Construct Toeplitz matrix by shifting elements
    for i ← 0 to m − 1 do
        matrix[i] ← toeplitz[i : i + n]          ▷ Extract sliding window
    end for
    // Compute final hash using matrix multiplication
    result ← (matrix × key) mod 2
    return result
end function
```

Implementation Details

The Toeplitz matrix construction and application process follows these steps:

1. **Matrix Dimension Calculation**:
   - Input dimension: n (original key length)
   - Output dimension: $m = n -$ security_parameter
   - Matrix size: $m \times n$.
2. **Toeplitz Matrix Generation**:
   - Generate random sequence: $t \in \{0, 1\}^{n+m-1}$
   - Construct matrix $T_{m \times n}$ where each row is a shifted subset of t
   - Matrix elements: $T_{i,j} = t_{i+j}$ for $0 \leq i < m, 0 \leq j < n$.
3. **Key Compression**:
   - Final key calculation: $k_{final} = T \cdot k_{input} \bmod 2$
   - Resulting key length: $m < n$.

Security Considerations

The Toeplitz matrix construction provides several security guarantees:

- **Uniformity**: The output bits are uniformly distributed
- **Compression Ratio**: The security parameter ensures sufficient compression:

$$\text{compression\_ratio} = \frac{m}{n} = 1 - \frac{\text{security\_parameter}}{n} \tag{23}$$

- **Error Propagation**: Single bit errors in input propagate unpredictably in output

Matrix Properties

The constructed Toeplitz matrix exhibits specific properties:

1. **Constant Diagonals**: Elements along each diagonal are identical
2. **Binary Elements**: All matrix elements $\in \{0, 1\}$
3. **Dimensions**:
   - Rows: $m = n - \text{security\_parameter}$
   - Columns: n (original key length)
4. **Structure**: For $i, j \geq 0$:

$$T_{i+1,j+1} = T_{i,j} \tag{24}$$

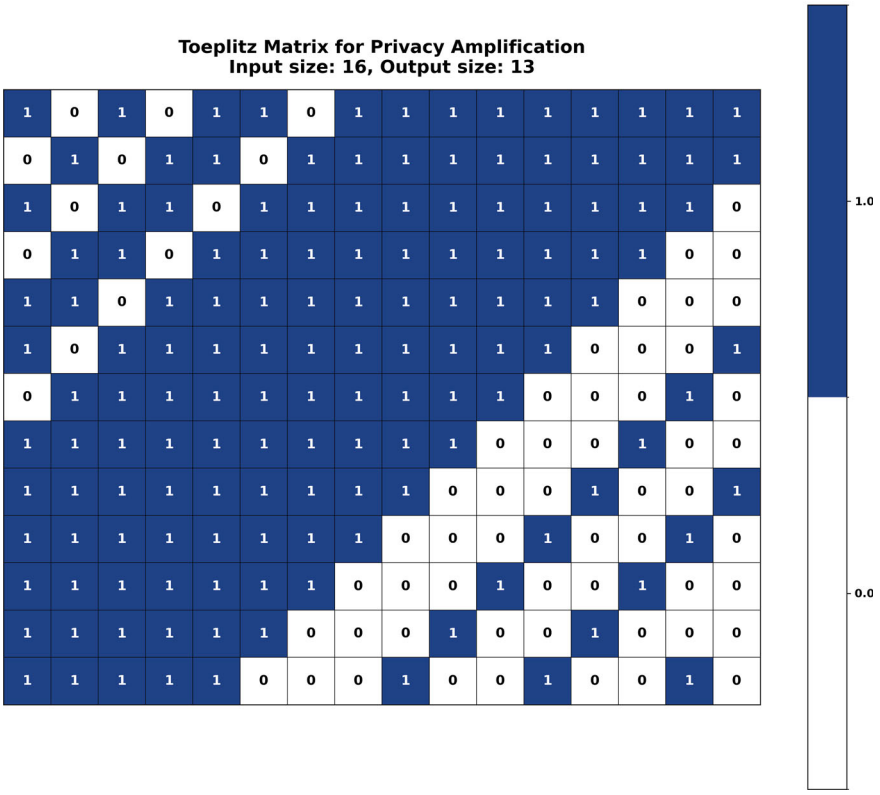


Fig. 3 Toeplitz matrix structure (n = 16, m = 13) showing the binary pattern

This implementation ensures both computational efficiency and security in the final key generation process, with the visualization in Fig. 3 demonstrating the matrix structure used in the actual implementation.

Performance Metrics

Two primary metrics are used to evaluate the protocol performance:

- Key rate (R): The ratio of final secure key length to initial qubit number
- Holevo bound (B): Theoretical upper bound on the information accessible to an eavesdropper.

Error Correction in QKD

After the quantum transmission, Alice and Bob must correct errors in their keys. A simple model for error correction is given as:

$$K'_B = f(K_A, K_B) \tag{25}$$

where K_A and K_B are the initial keys of Alice and Bob, respectively, and f is the error correction function [11].

Key Rate in QKD

The key rate R is the ratio of the final key length to the initial key length:

$$R = \frac{|K''|}{n} \tag{26}$$

where $|K''|$ is the length of the final key after privacy amplification and n is the number of qubits transmitted [5].

Holevo Bound

The Holevo bound provides the upper limit on the amount of classical information that an eavesdropper (Eve) can obtain about the quantum communication between Alice and Bob:

$$\chi = S(\rho_E) - \sum_i p_i S(\rho_E^i) \tag{27}$$

where $S(\rho)$ is the von Neumann entropy of the state ρ, ρ_E is Eve's quantum state after the interaction, ρ_E^i are the possible states of Eve's system conditioned on Alice's choice, and p_i is the probability of each state [12].

3 Results

This section provides clear information and offers a deeper insight into the performance of our simulation under different noise conditions, key lengths, and error rates. The key rate is used as the performance metric. An optimal point is determined for each of the models, where a soft spot is obtained for the different parameter combinations. We summarized Figs. 4, 5a, b, 6a, b and prepared a table (see Table 3) where we compared the different noise models' optimal points alongside their key characteristics.

Research Observations

The analysis of different noise models revealed several key findings:

- Distinct optimal points for each noise model
- Key length and error rate impact varies significantly

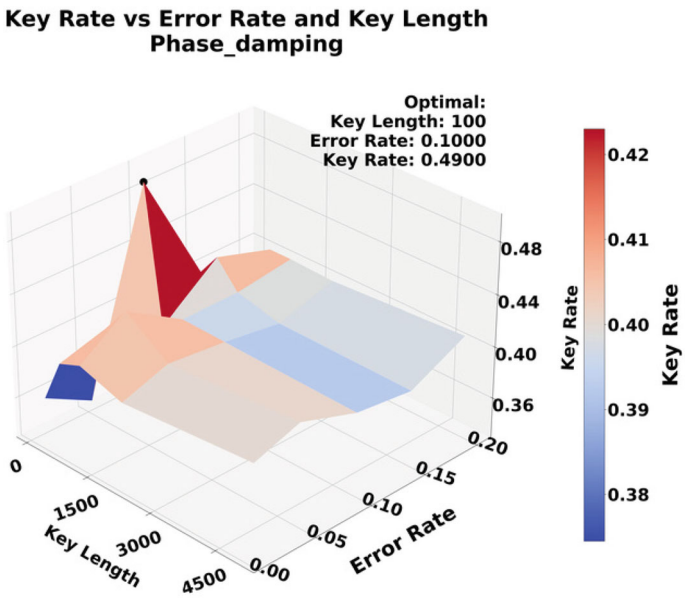
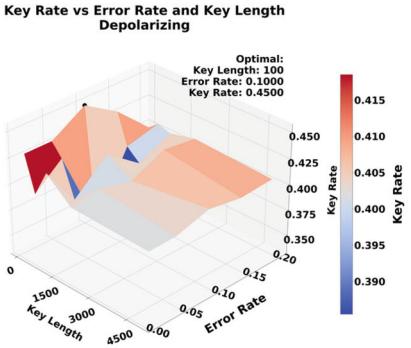
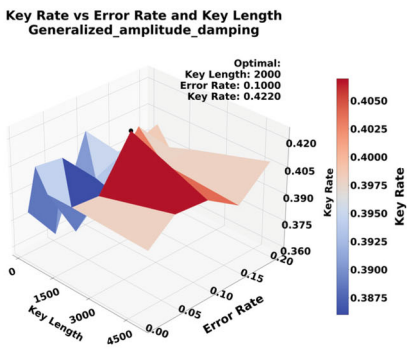


Fig. 4 QKD simulation results using 3D heat map showing phase damping effects

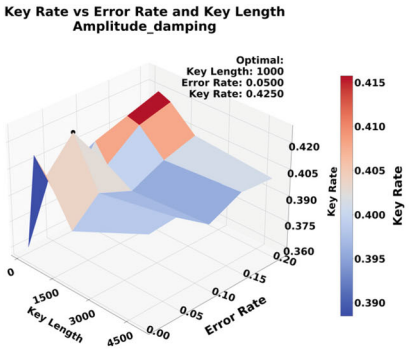


(a) Depolarizing channel effects

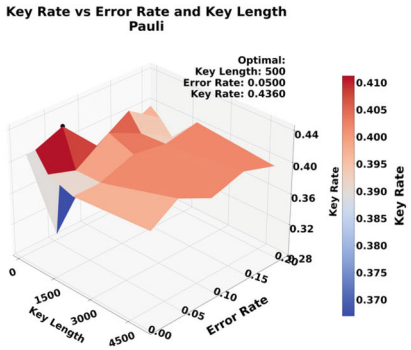


(b) Generalized amplitude damping effects

Fig. 5 QKD simulation results using 3D heat maps comparing different noise channels



(a) Amplitude damping effects



(b) Pauli channel effects

Fig. 6 QKD simulation results using 3D heat maps for additional noise models

- Higher error rates don't always lower key rates
- Longer key lengths not universally better
- Performance highly dependent on noise type
- Optimization needs both key length and error rate consideration.

Key Rate Versus Measured Error Rate Analysis

We analyzed the key rate versus the measured error rate with Holevo bound (see Fig. 7), and made the following key experimental observations:

- Most data points cluster below the Holevo Bound, indicating good performance across noise models
- Error rates primarily range from 0.1 to 0.4, with key rates between 0.3 and 0.5

Table 3 Comparison of different noise models and their performance characteristics

Noise model	Optimal point	Key characteristics
Depolarizing	0.450 (0.1, 100)	Consistent across key lengths; slight decrease with error rate; robust performance
Amplitude damping	0.425 (0.05, 1000)	Varies with key length; better at longer lengths for high errors; sensitive to changes
Phase damping	0.490 (0.1, 100)	Highest variability; decreases with key length; unique behavior
Generalized amplitude damping	0.422 (0.1, 2000)	Consistent across conditions; no clear trend; resilient to parameter changes
Pauli noise	0.436 (0.05, 500)	Highest performance range (0.290–0.436); improves with key length; most sensitive to changes

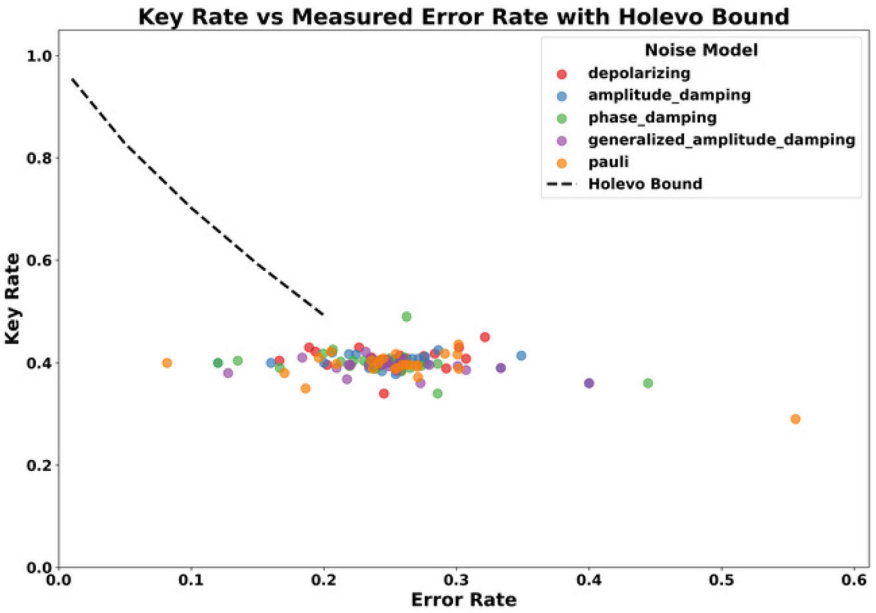


Fig. 7 Scatter plot of key rate versus measured error rate showing relationship to the Holevo bound

- A few outliers exist, particularly for the Pauli noise model at higher error rates.

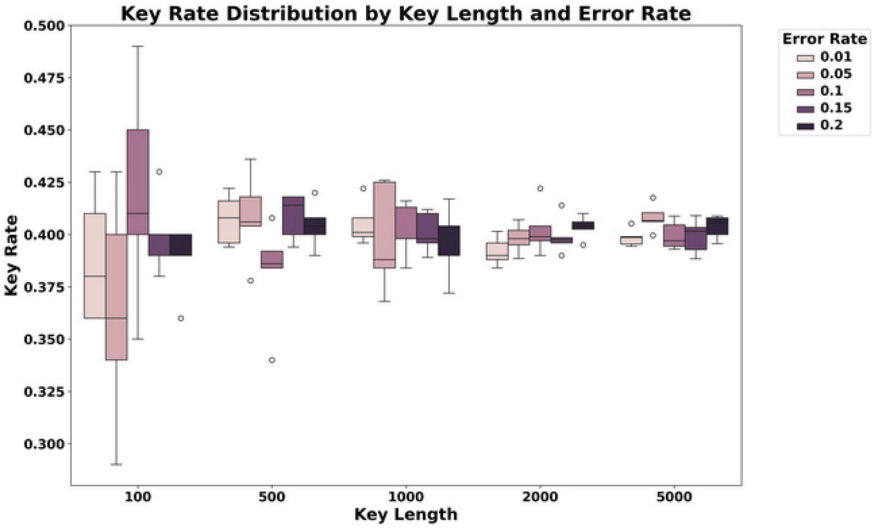


Fig. 8 Box plot analysis of key rate distributions across different configurations

Table 4 Detailed analysis of key rate distributions across different error ranges and key lengths

Key length	Low error (0.01–0.05)	Mid error (0.1)	High error (0.15–0.2)
100	High variability (0.29–0.43). Widest distribution	Highest peak performance (0.49). Large spread	More consistent performance. Narrower distribution
500	Moderate variability. Improved consistency from 100	Consistent performance. Slight improvement in median	Slight improvement in consistency. Fewer outliers
1000	Lower variability. Narrower distribution than 500	Consistent performance. Similar to 500 but more compact	Consistent performance. Narrow distribution
2000	Consistent performance. Narrow distribution	Slight improvement in median and consistency	Very consistent performance. Compact distribution
5000	Most consistent across all lengths. Very narrow distribution	Most consistent. Highest median for this error range	Most consistent. Narrow distribution with high median

3.1 Key Rate Distribution Analysis

Figure 8 illustrates the relationship between key rate distribution, key length, and error rate. The observations are detailed in Table 4. From the data, several notable insights can be inferred, as the quantum simulation reveals distinct performance patterns across various noise landscapes.

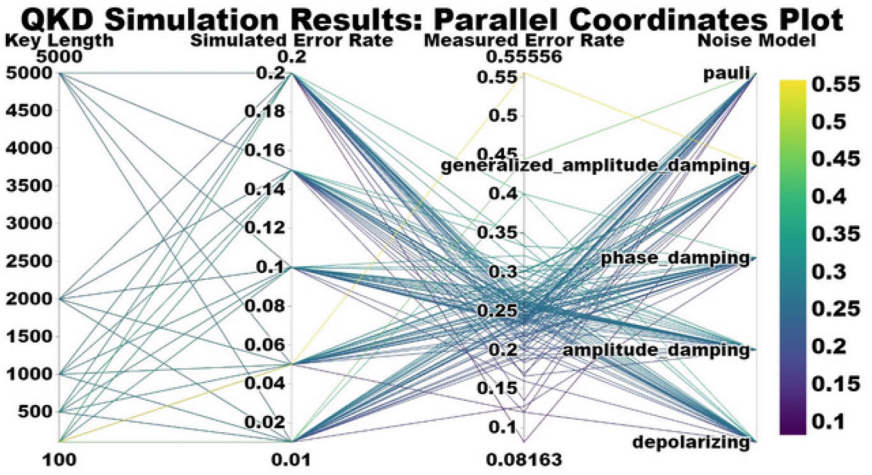


Fig. 9 Parallel coordinate plot showing relationships between key parameters across different noise models

3.2 QKD Simulation Results Using Parallel Coordinate Plot

The parallel coordinate analysis revealed several key characteristics across the noise models. To begin with, the Pauli noise model showed the highest inconsistency coupled with the largest measured error rates among all the examined models. In contrast, the depolarizing noise showed commendable consistency when simulated across different key lengths. In the case of phase damping, it showed noticeable sensitivity to changes in both key length and error rate. The generalized amplitude damping model displayed moderate behavior across most measured parameters, placing it between the extremities observed in other models. These distinguishing patterns provide valuable insights into the relative stability and impact of different quantum noise channels on the QKD protocol performance. Figure 9 illustrates a parallel coordinate plot that highlights the relationships between key parameters across different noise models.

4 Conclusions and Future Work

Our analysis of BB84 quantum key distribution, incorporating error correction and privacy amplification systems under various noise models has led to several important conclusions. The analysis demonstrates that BB84 QKD system performance is deeply dependent on the specific noise model affecting the quantum channel. This finding highlights that there cannot be a ubiquitous approach to system optimization, as different noise environments require distinct strategies. As a result, the accurate

characterization of the quantum channel emerges as a critical prerequisite for implementing effective BB84 QKD systems, since it directly influences both the choice of protocol parameters and the achievable secure key rates.

4.1 Future Directions

Several promising directions for future research emerge from this interesting study. The optimization potential of the current protocol implementation calls for further investigation to enhance performance. Additionally, implementing the E91 protocol would provide valuable comparative insights into protocol resilience under various noise conditions. An in-depth investigation of finite-key effects would bridge the gap between theoretical bounds and practical implementations. Furthermore, the development of adaptive protocols capable of responding to mixed noise environments represents a crucial step toward more robust real-world QKD systems. These research directions collectively aim to advance the practical implementation of quantum key distribution in realistic conditions for BB84 in particular, and by extension all other QKD protocols.

References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, Cambridge (2010)
2. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, pp. 175–179 (1984)
3. Renner, R.: Security of quantum key distribution. International Journal of Quantum Information **6**(1), 1–127 (2008)
4. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10th anniversary edition edn. Cambridge University Press, Cambridge, UK (2010)
5. Scarani, V., Bechmann-Pasquinucci, H., Cerf, N.J., Dušek, M., Lütkenhaus, N., Peev, M.: The security of practical quantum key distribution. Reviews of Modern Physics **81**(3), 1301–1350 (2009) https://doi.org/10.1103/RevModPhys.81.1301
6. Winderl, D., Franco, N., Lorenz, J.M.: Quantum neural networks under depolarization noise: Exploring white-box attacks and defenses. Quantum Machine Intelligence **6**(2), 83 (2024)
7. Banaszek, K.: Quantum operations and their applications in quantum communication. Physical Review A **69**(1), 1–6 (2004) https://doi.org/10.1103/PhysRevA.69.010304
8. Breuer, H.-P., Petruccione, F.: The Theory of Open Quantum Systems. Oxford University Press, Oxford, UK (2002)
9. Zhan, C., Yang, J., Hu, G., Zhang, Y., Guo, S., Zhou, H.: Expanded nested mimo radar for dod and doa estimation based on tensor model. In: 2024 4th International Conference on Information Communication and Software Engineering (ICICSE), pp. 120–125 (2024). https://doi.org/10.1109/ICICSE61805.2024.10625683
10. Bennett, C.H., Brassard, G., Crépeau, C., Maurer, U.M.: Generalized privacy amplification. IEEE Transactions on Information Theory **41**(6), 1915–1923 (1995) https://doi.org/10.1109/18.476316

11. Gottesman, D., Lo, H.-K., Lütkenhaus, N., Preskill, J.: Security of quantum key distribution with imperfect devices. IEEE Transactions on Information Theory **49**(9), 457–475 (2004) https://doi.org/10.1109/TIT.2004.833360
12. Holevo, A.S.: Bounds for the quantity of information transmitted by a quantum communication channel. Problemy Peredachi Informatsii **9**(3), 3–11 (1973)

Benchmarking ML, DL, and Transformer Models for Document Classification

Ricardo Rioda Santiago, Antonio Contreras Ortiz, Claudia Tona, and Miguel Lopez-Montiel

Abstract This study benchmarks the performance of machine learning (ML), deep learning (DL), and transformer models for binary classification of text and images from documents, focusing on curriculum vitae (CVs) and research papers. The primary contributions are the development of two pipelines: one for text extraction and another for training 24 classifiers across 4 datasets, along with a performance comparison of classifiers for both textual and visual data. Our findings reveal that traditional ML models often outperform DL models for text classification due to faster inference times while maintaining adequate accuracy. Conversely, pre-trained models excel in image classification, outperforming non-pre-trained models. While text classifiers demonstrate robust generalization to unseen data, image classifiers face challenges. We also propose strategies based on computational resources and data types for classifying CVs and research papers. Our analysis shows that ML models on CPUs deliver rapid inference, whereas DL and Transformer models on GPUs can be up to 645× slower for text tasks and 562× slower for image tasks. Despite this, DL and Transformer models achieve up to 43% higher F1-scores in complex scenarios, particularly for image classification. This work explores the trade-offs between model complexity and performance, guiding model selection for specific use cases.

Keywords Deep learning · Document classification · Vision transformer · BERT · Transformers · Machine learning

R. Rioda Santiago · A. Contreras Ortiz · M. Lopez-Montiel (✉)
ITJ Labs, Tijuana, Baja California, Mexico
e-mail: miguel.lopez@itj.com

R. Rioda Santiago
e-mail: ricardo.santiago@itj.com

A. Contreras Ortiz
e-mail: jose.contreras@itj.com

C. Tona
Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California, Parque Internacional Industrial Tijuana, Tijuana, Baja California, Mexico
e-mail: tona.claudia@uabc.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_18

1 Introduction

Document classification is essential to improve information processing, searching, and retrieval. It streamlines workflows by enabling rapid identification, classification, and labeling of unstructured data. In recent years, numerous new algorithms have been developed in the field of deep learning. Notably, the introduction of transformers [1], an architecture that has outperformed recurrent neural networks (RNNs) on various natural language processing (NLP) tasks, has revolutionized natural language understanding and processing, becoming the standard approach for these tasks. Transformers are revolutionary as they use a self-attention mechanism, which allows the model to generate a representation of a statement by weighing the significance of various terms in the sentence [2]. This feature makes transformers more effective than RNNs in handling context and long-range dependencies, which makes them especially useful for applications like sentiment analysis, text summarization, and machine translation. Transformers were first created for machine translation, but their popularity has spread to many other NLP domains and beyond. Hugging Face, an important deep learning community in which individuals actively shared their findings and pre-trained models, was also made possible by this development, greatly expanding the field's research [3]. This collaborative environment encouraged further exploration of the transformer mechanism, expanding its application beyond natural language processing into computer vision. The integration of vision transformers (ViT) with convolutional models (CNNs) further enhanced performance by combining the strengths of both techniques. While each architecture has its own strengths and limitations, their combined approach often yields superior results across a range of tasks, highlighting the importance of blending different technologies to achieve more efficient and accurate solutions in computer vision [4, 5]. This synergy is particularly evident in benchmark evaluations such as the RVL-CDIP dataset [6], which comprises nearly 400,000 scanned document images categorized into 16 classes, including letters, forms, emails, resumes, and memos. The top five solutions for this dataset incorporate ViT, reflecting their superior performance compared to purely convolutional models. Even among the top 20 solutions, many still rely on convolutional architectures, highlighting the continued relevance of traditional approaches. But does everyone really need to use these super complex architectures and models to solve their tasks? Are those incorporating advanced architectures like ViT overly expensive or excessive for certain applications? These questions have inspired the present study, which aims to explore these practical implications by benchmarking various models. By evaluating different architectures, seeking to determine the trade-offs between model complexity and performance. Based on the previous questions, we propose the following hypothesis (H_0): *using pre-trained models such as transformers and advanced architectures for classifying documents such as resumes and scientific articles in image or text format do not generate a significant difference in the accuracy of the models compared to traditional machine learning models, resulting in excessive use of computational resources.* To test our hypothesis, we develop two binary classification tasks: identifying whether a document is a curriculum vitae

(CV) and determining if a document is a scientific paper. These tasks are essential for improving document processing, search, and retrieval in human resources and academic research. In this work, when we mention curriculum vitae, resume, or CV, we refer to the same entity. This paper presents the following contributions:

- Proposed pipelines: We introduce two pipelines: one for text extraction from documents and another for evaluating and training classifiers. Four datasets (two per task) were developed, containing image or text data.
- Comparative analysis of models: We investigate whether traditional machine learning models outperform deep learning and transformer models when large training datasets are unavailable.
- Pre-trained versus non-pre-trained models: The study examines whether pre-trained models offer superior performance over non-pre-trained models for visual data classification.
- Text versus image-based classifiers: We assess whether text-based classifiers are more effective than image-based classifiers in generalizing document classification tasks.
- Optimal strategy for document classification: We provide insights into the best approach for classifying CVs and scientific articles, balancing computational efficiency and the type of input data used.

By conducting these experiments, we aim to provide insight into the practical performance and applicability of these models in real-world scenarios. Our findings will influence future research directions and industry practices by clarifying whether the complexity of advanced architectures like transformers is warranted for everyday tasks or if simpler models can deliver comparable performance with reduced computational costs. This could lead to more efficient and cost-effective approaches in both research and industry, driving innovation and optimizing resource use in ML applications. The structure of this paper is organized as follows: Sect. 2 outlines the fundamental concepts underlying the models used in this study. Section 3 reviews the relevant literature on document classification techniques, focusing on ML, deep learning, and transformer models. Section 4 details the methodology, including data collection, preprocessing, and model training procedures. Section 5 presents the experiments conducted, the results obtained from the various models, and a discussion comparing these results with existing benchmarks and related studies. Finally, Sect. 6 summarizes the conclusions and suggests directions for future work.

2 Fundamentals

In this section, we will introduce the foundational concepts and terminologies essential to understanding the techniques and methodologies discussed in this paper. We begin by exploring the core principles of machine learning (ML) and artificial intelligence. Following that, we will examine how to select the appropriate model based

on factors such as data type, computational resources, budget, and desired outcomes. After establishing the fundamentals, we will focus on specific domains of application. Computer Vision and Natural Language Processing are two prominent subfields that demonstrate the power and versatility of ML techniques.

2.1 Machine Learning

Machine learning, a branch of artificial intelligence, enables computers to learn patterns, identify trends, and make decisions without explicit programming. Although these techniques were first developed in the 20th century, their widespread application in fields such as computer science and medicine did not occur until much later. The delay was mainly due to hardware limitations. Machine learning was computationally expensive, and the CPUs of that era lacked the resources, resulting in long training times even for small datasets. This contributed to a decline in interest and research in the field, a period known as the AI winter [7]. Interest in ML was revived with the development of the back-propagation algorithm [8], which offered an efficient method to train neural networks. This algorithm was a significant milestone in the evolution of deep learning and ML, as it addressed the challenge of long training times, enabling researchers to train deep neural networks more efficiently. At the same time, advances in video game cards, originally designed for fast pixel rendering, were soon repurposed by researchers for broader computational tasks. Recognizing this potential, NVIDIA began developing general-purpose graphics processing units (GPUs) and introduced the CUDA interface, an extension of C, to facilitate programming on GPUs [9]. One of the key features of GPUs is their capacity to run multiple operations in parallel, significantly reducing processing times. This capability greatly benefited artificial intelligence applications, leading to substantial reductions in training times.

2.2 Prerequisites for Learning Types

When selecting a machine learning approach, it is essential to consider several key criteria, including the type of data, the complexity and size of the dataset, the computational resources, and the desired outcomes. These factors influence the choice of learning algorithms and system design. One of the primary considerations is the type of data involved, which can be categorized as structured or unstructured, and further into labeled or unlabeled data:

- **Structured Data**: This type of data is composed of observations, also known as records, and features or fields, typically arranged in a hierarchical or row-column structure, where each observation is a record and each feature is a specific attribute of that record. A typical example is a medical records database, where each patient

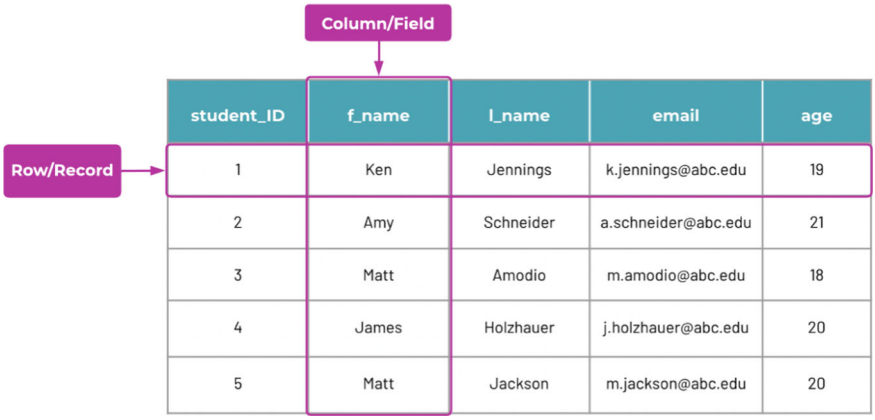


Fig. 1 Structure data example. This figure shows a database table as an example of structured data

has data like age, weight, medical conditions, and last hospital visit. An example of this can be seen in Fig. 1.

- **Unstructured Data**: This type of data lacks a specific order and isn't typically organized in rows and columns. Examples include images, audio, text, and video. As Sinan Ozdemir mentions in Principles of Data Science, "Most estimates place unstructured data as 80–90% of the world's data" [10]. However, as previously mentioned, with proper cleaning and transformation, any data can be adapted to fit any model. Examples of unstructured data in the Fig. 2:

In addition to the type of data, other criteria must be considered:

- **Dataset Complexity and size**: Many algorithms are designed to detect complex patterns in data; some cannot effectively handle non-linear data. On the other hand, poor data cleaning can degrade performance, and lack of regularization can lead to over-fitting.
- **Computational Resources**: The choice of model often depends on available computational resources and budget. While CPUs are common, they are typically insufficient for training deep learning models, which require GPUs to manage the high computational demands and reduce training time. For instance, training a CNN on a CPU is generally impractical due to its complexity.

2.3 Computer Vision

Computer vision is a branch of artificial intelligence that gives computers the ability to "see," or interpret, analyze, and visualize data from many sources, including photos and videos, simulating human vision to aid in process automation. The collection of technologies emerging from this domain has found wide use, ranging from

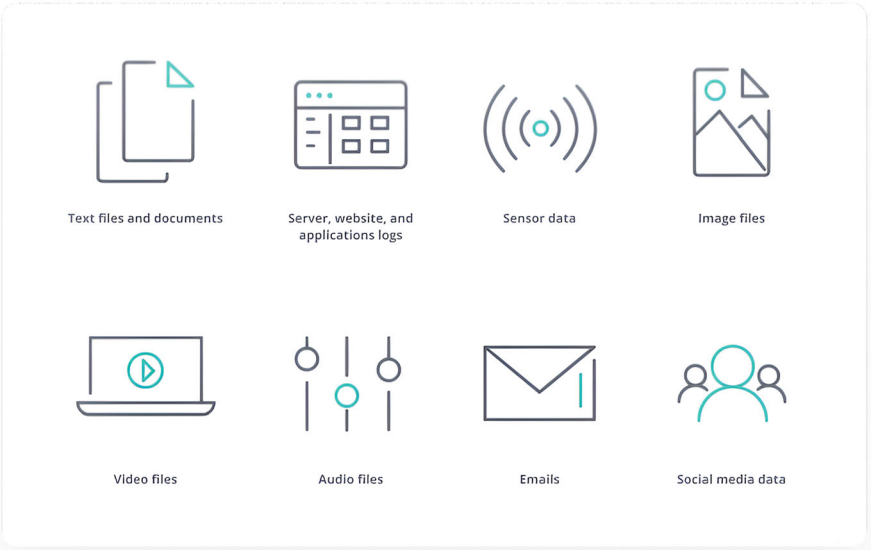


Fig. 2 Examples of unstructured data

autonomous driving cars to the detection of industrial defects [11]. From that collection emerged the well-known convolutional neural networks, one of the main components of the field, excelling in tasks including object detection and image classification since they automatically extract hierarchical information from images [7]. Vision Transformers (ViT) represent a newer approach, utilizing self-attention mechanisms from natural language processing [2] to capture relationships within images, producing groundbreaking results in a variety of benchmarks, including the RVL-CDIP dataset [12]. Although CNNs remain highly effective, especially in real-time tasks like object detection, where models like YOLO dominate [13], emerging transformer-based models, such as Detection Transformer (DETR) [14] and real-time DETR (RT-DERT), are showing competitive performance, offering new alternatives.

2.4 Natural Language Preprocessing

A branch of artificial intelligence called natural language processing (NLP) provides computers with the ability to understand, interpret, and produce human language [15]. Originating in the 1950s, NLP began with basic translations and rule-based models but faced limitations due to language complexity. The 1980s introduced statistical methods and recurrent neural networks (RNNs) [16], leading to improvements in language processing. Significant advancements occurred between 2000 and 2010 with

word embeddings and sequence-to-sequence architectures, which improved translation accuracy and efficiency. The transformer architecture, introduced in 2017, revolutionized NLP by improving text processing and handling long-range dependencies through attention mechanisms [1].

NLP encompasses various tasks, such as text classification, sentiment analysis, Natural Language Understanding (NLU), text summarization, text generation, and question answering. Sentiment analysis, for example, classifies text to gauge product acceptance based on reviews. NLU supports advanced applications by enabling text summarization, generation, and translation. Traditionally, methods such as decision trees and logistic regression addressed NLP challenges, but transformers are now the industry leader with their ability to manage complex language patterns and contextual information [2].

3 Related Work

When discussing model selection, a crucial aspect of any regression or classification task, its relative importance is often debated. However, as highlighted by Google in their renowned study, 'The Unreasonable Effectiveness of Data' [17], prioritizing the quantity and quality of data is essential before focusing on model considerations. In document classification, the data usually consists of a mix of textual information, images, and PDFs. These documents can vary significantly in quality, ranging from clear, well-scanned documents to blurry or poorly formatted ones. If the data is not of sufficient quality, even the most advanced models may struggle to accurately classify the documents. The study by Sun et al. [18] reinforces this notion by demonstrating that improvements in model architecture do not necessarily translate into better performance if the data quality is not addressed first. Moreover, Dodge et al. [19] emphasize that most vision systems are trained on high-quality data and are generally susceptible to blur and noise in images, which significantly impacts the performance of most neural network architectures. However, since we cannot guarantee that users will upload high-quality data, it is necessary to use data representative of what people might actually submit. Once data quality and sufficiency are established, the next step is to select an appropriate model. For tasks like computer vision or natural language processing, choosing the right model is crucial. In the field of computer vision, deep learning architectures are often preferred over traditional ML models for several reasons. For instance, a 2021 study by Wang et al. [20] compares the performance of a support vector machine (SVM) and a convolutional neural network (CNN) across four different image classification datasets. The study found that while SVMs were easier to implement and generally had lower training times compared to CNNs, their accuracy often fell short of that achieved by CNNs. Furthermore, SVMs seemed to reach a performance plateau, unable to surpass the accuracy levels of CNNs in most cases. For document image classification on the RVL-CDIP dataset, current benchmark results highlight the dominance of Vision

Transformer models and CNN-based architectures. The leaderboard showcases top-performing models, such as EAML [21], which achieved 97.70% accuracy in 2021, and other high-ranking models that frequently incorporate attention mechanisms, like DocFormer (2021) and LayoutLMv3 (2022) [22, 23]. These results underscore the trend toward using advanced attention-based architectures to achieve state-of-the-art performance in document classification tasks. In natural language processing and its subfields, current state-of-the-art approaches are generally categorized into two main types: generative and discriminative models, as detailed in [24]. Generative models, such as large language models like ChatGPT, BERT, GEMINI, and LLAMA, generate synthetic data by modeling probability distributions. In contrast, discriminative models, such as logistic regression and conditional random fields (CRFs), predict probabilities based on observed examples. Generative models, while offering high performance, often require substantial computational resources and are more resource-intensive to train from scratch. However, if pretrained on related tasks, their training can be more efficient. Discriminative models, in contrast, generally train more quickly and with less computational cost. Despite the higher computational cost, generative models tend to achieve greater accuracy and performance metrics compared to discriminative models. For instance, a 2023 study [25] demonstrated that the BERT model consistently outperformed traditional discriminative models across four datasets (two in English, one in Portuguese, and one in Chinese) by at least 7% in accuracy. However, it's worth noting that training the BERT model was more computationally demanding compared to the traditional ML models. When faced with a task that can be tackled using either computer vision or natural language processing, several key questions arise: Which approach is more suitable? Which will yield better results? Should we invest in pre-training a complex neural network, or can simpler models be enough? To answer these questions, we first need to consider the nature of the data. How complex is it? Is the vocabulary of the documents or the features of the images consistent or variable over time? Can we expect uniform quality from users interacting with the model? Additionally, are we aiming for predictions within shorter timeframes or better predictions? These factors will guide our decision on whether to use computer vision, natural language processing, or a combination of both, and help determine the appropriate model complexity and training strategy. In the literature, specific criteria for choosing between complex and simple models have been proposed. Studies suggest simpler models are often preferred for tasks requiring high interpretability, low computational resources, and stable data. For example, models like logistic regression or decision trees are beneficial in scenarios needing clear decision boundaries and transparency. In contrast, complex models like deep neural networks are essential for large-scale or highly variable data. For example, CNNs handle intricate image recognition, while transformer models like BERT excel in sophisticated natural language understanding. These models, while more computationally intensive, can capture complex patterns that simpler models might miss. The choice of model should balance data complexity, task requirements, and available resources to find the most effective approach.

4 Methodology

In this work, we designed two binary classification tasks: distinguishing between resumes and non-resumes, and between scientific papers and non-scientific papers. Our objective was to compare the performance of traditional machine learning, deep learning, and transformer models (including large language models and vision transformers) using text versus image inputs to determine the most effective approach [26]. These tasks are critical in fields such as human resources management and academic research. Accurate classification can streamline document processing, improve search functionality, and improve information retrieval systems. These tasks test the models' ability to discern subtle differences in structure and content, making them ideal for evaluating the effectiveness of various text and image-based classification approaches [27].

Figure 3 illustrates that for each classification task, there are two datasets: a text dataset and an image dataset. We will create models based on machine learning, deep learning, and transformers and finally compare their performance using various metrics. We have defined a pipeline that evaluates the performance of classification models using text and images as input for their training.

In Fig. 4 we can observe the classification pipeline used to evaluate performance:

1. We collect the internal raw data from ITJ and other sources.
2. For each task, there are two datasets (text and image data sets).
3. Each dataset is split into training, validation, and test sets.
4. Preprocessing the data is necessary to reduce model complexity.

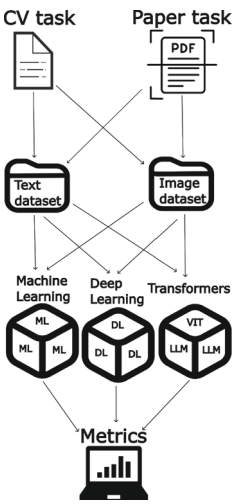


Fig. 3 Binary classification tasks. This diagram shows the flow of each task to obtain the metrics

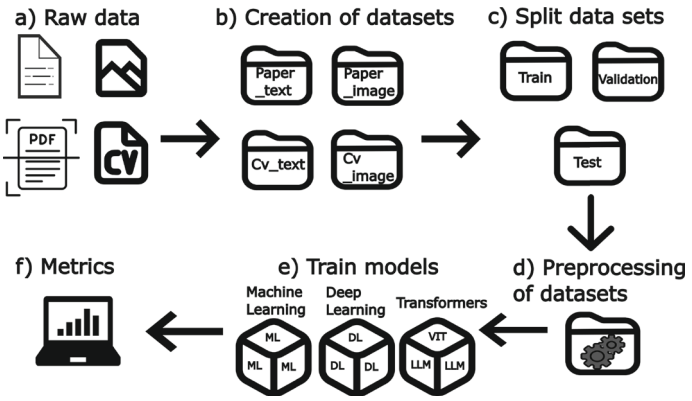


Fig. 4 Classification pipeline. The pipeline shows the steps applied to each task to train their models and finally perform comparative analysis

5. Classifiers based on machine learning (ML), deep learning (DL), and transformers are created using the datasets.
6. A comparative analysis is conducted using time inference, precision, recall, and F1-score.

4.1 Classification Tasks

The vision-based and text-based approaches involve two independent tasks: determining whether a document is a resume and whether it is a scientific paper. The goal of these tasks is to compare models based on their respective input data, using text or images for the resume and scientific paper classification tasks, respectively.

4.2 Data Collection

We used a private dataset that is not publicly available. This dataset includes professional resumes detailing individuals' work experience, education, skills, and achievements. It also contains business plans, market analysis reports, financial statements, contracts, agreements, and other legal texts. Due to the inclusion of personal information, this dataset cannot be released.

The dataset includes a total of 15,000 files, of which approximately 10,000 are curriculum vitae from actual applicants:

- Only PDF, Word, and image files were selected for gathering raw data for both curriculum vitae information and random data classes.
- Additionally, scientific papers were collected from files available on arXiv.

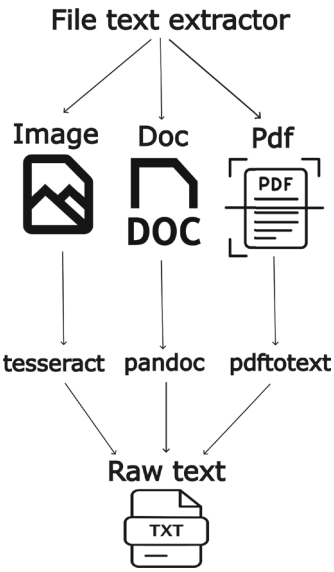


Fig. 5 Text extractor pipeline. This diagram shows that, depending on the type of file, the tool that will be used to extract the text from the file

The only public datasets used in our research are the following: The resume_corpus dataset contains about 29,000 resume contents [28]. RVL-CDIP dataset [6], which comprises nearly 400,000 scanned document images classified into 16 classes, including letters, forms, emails, resumes, and memos.

4.2.1 Data Extraction from Files

Before creating the image and text datasets, we collect the information we consider relevant, such as text or images from files and datasets. For image extraction, we use a Python script that extracts the first page from a PDF file using a library called pdf2image. For scientific articles and CVs, the first page holds the most critical information. In scientific articles, this typically includes the title and abstract. We only process PDF, Word, and image files for text extraction. Our pipeline identifies the file type and selects the appropriate tool for information extraction. We selected each specific tool because they can be easily used in a bash script. Since our entire pipeline is run from a bash script, it is responsible for identifying the file type and then using the correct tool. Because some PDF files were scanned, it is impossible to extract text directly from them because they are images converted to PDF.

As we can see in Fig. 5, to perform text extraction, we first identify the type of file. In our case, it can be an image, a Word file, or a PDF. Once the file type is identified, the corresponding tool is used.Text is extracted from images using Tesseract,[1] from Word

[1] Tesseract is an open-source open-source optical character recognition (OCR) engine, available at: https://github.com/tesseract-ocr/tesseract.

files using Pandoc,[2] and from PDFs using pdftotext.[3] Finally, the raw text is extracted from the original source material, which may include documents, transcripts, or other forms of written content.

4.3 Text Preprocessing

Cleaning and reducing noise in the data before training helps improve model performance. The following steps outline the process:

1. **Remove special characters**: Standardize text by removing accent marks, reducing variability without altering meaning.
2. **Remove digits**: Reduces noise and dimensionality by eliminating numbers that are often irrelevant to text semantics.
3. **Remove HTML tags**: Focus on content by eliminating formatting information.
4. **Remove URLs**: Eliminates irrelevant web addresses, reducing the noise in the text.
5. **Remove stop words**: Reduces vocabulary size, allowing the model to focus on more informative words.

4.4 Image Preprocessing

The preparation of images for training involves several pre-processing steps to ensure that the images are suitable for feeding into a machine learning model. The following steps outline the process:

1. **Convert to gray-scale**: Simplifies the image data, reducing computational load, and focusing on intensity.
2. **Apply thresholding/binarization**: Highlights key features by converting the image to binary, reducing complexity.
3. **Straighten/Align**: Aligns text or features correctly, improving the accuracy of feature extraction.
4. **Resize to maintain aspect ratio**: Ensures uniform image dimensions while preserving proportions.
5. **Pad to maintain target size**: Conforms to required input dimensions, maintaining content integrity.

[2] Pandoc is a universal document converter capable of converting files between different formats, available at: https://pandoc.org/.

[3] pdftotext is a command-line tool to extract text from PDF files, available at: https://www.xpdfreader.com/pdftotext-man.html.

Table 1 Datasets summary. This table shows the distribution of data sets and classes per data set

Dataset	Training samples	Validation samples	Test samples	Classes
Cv_text	7,208	2,312	2,332	2
Cv_image	8,924	1,192	1,784	2
Paper_text	7,208	2,312	2,332	2
Paper_image	8,924	1,192	1,784	2

4.5 Image/Text Datasets

The first text-based dataset contains approximately 12,000 documents, evenly split between CVs and random documents. The second text-based dataset also has around 12,000 documents, divided equally between scientific papers and random documents. For image-based data, we created two additional datasets of about 11,000 images each, one divided between CVs and random documents, and the other between scientific papers and random documents.

The Table 1 presents the datasets, which have been divided into training, validation and test sets for two tasks: CV and scientific article. Data selection was performed using stratified sampling to ensure balanced classes in both tasks.

4.6 Machine Learning Models

In this section, we provide a brief overview of the process followed before feeding data into the machine learning models and describe the machine learning models used for our benchmark. For image classification, the general preprocessing steps outlined in Sect. 4.4 were applied, but an additional step was necessary: flattening the images to convert them into suitable data vectors for the algorithms. In the case of text, the general steps of preprocessing text were mentioned in Sect. 4.3, but before feeding to machine learning algorithms, we created a simple pipeline consisting of 3 steps:

1. **CountVectorizer**: This transforms the text data into a matrix of token counts.
2. **TfidfTransformer**: This converts the count matrix to a normalized Term Frequency-Inverse Document Frequency (TF-IDF) representation.
3. **SGDClassifier/LogisticRegression**: After the two steps before, we feed any of the two selected models.

4.6.1 SGDClassifier

The stochastic gradient descent (SGD) classifier is a linear classifier optimized with stochastic gradient descent. The SGD Classifier is versatile and can be configured with different loss functions, such as hinge loss for linear Support Vector Machines (SVMs) or log loss for logistic regression. This model is used for both image and text classification [29]. For the training of this model, we used the following hyperparameters: We employed hinge loss, a robust and effective choice for linear Support Vector Machines (SVMs). To avoid over-fitting, the L2 regularization was applied, penalizing large coefficients. The regularization term (alpha) was set to 0.001, balancing the trade-off between data fitting and regularization. The maximum number of iterations was set to 5, which was sufficient for convergence in our experiments. The tolerance for stopping criteria was set to None, ensuring that the model ran for the specified number of iterations.

4.6.2 LogisticRegression

One of the most fundamental models for machine learning is called logistic regression; it consists of a linear model plus a sigmoid function, which enables the linear model to act as a classifier. In the context of this study, logistic regression was used to categorize text and image data into predefined categories. The preprocessing of the text data includes tokenization, stop word removal, and vectorization using methods such as TF-IDF to convert textual content into numerical features suitable for model training. Similarly, for the image data, pre-processing involved resizing, normalization, and feature extraction. After preprocessing, the logistic regression model was trained on these processed datasets, enabling it to identify relationships and trends in the data, which were then used to make predictions on unseen samples.

4.7 Deep Learning Models

In this section, we describe the deep learning models used for image and text classification. For image classification, the general preprocessing steps outlined in Sect. 4.4 were followed, with an additional step of flattening the images into suitable data vectors for the algorithms. For text classification, the preprocessing steps referenced in Sect. 4.3 were applied.

4.7.1 ConvNeXt V2 (Tiny-Sized Model)

ConvNeXt V2 is a convolutional neural network architecture designed for image classification tasks, featuring enhanced efficiency and accuracy. The model architecture and training process are detailed in [30]. In this study, we used the ConvNeXt V2

model, pre-trained with the FCMAE framework, and fine-tuned on the ImageNet-1K dataset at 224 × 224 resolution.

4.7.2 ResNet-50

Microsoft created the ResNet-50 model, a deep residual neural network architecture that was used for image classification tasks. The residual block, as its name suggests, is one of the novel features of the architecture, enabling the models to reduce both the partial reduction of the disappearing gradient issues and the performance loss of enormous architectures. In this study, we used the fine-tuned ResNet-50 model trained on the RVL-CDIP dataset. Detailed information on the model architecture can be found in [27].

4.7.3 TextCNN

TextCNN, a convolutional neural network specifically designed for text classification, is designed to capture the local dependencies of words in text data. The model architecture and training methodology are described in [31, 32]. TextCNN was used in this study to classify text data, using its capability to extract high-level features from text input.

4.7.4 CharCNN

Character-level Convolutional Neural Networks (CharCNN) operate on the character level, capturing subword information from text data. This model is particularly effective for languages with rich morphology. In this study, CharCNN was used to classify text according to character-level characteristics, as detailed in [33].

4.8 Transformer Models

In this section, we describe the transformer models utilized for image and text classification. For image classification, the general preprocessing steps described in Sect. 4.4 were followed, with the added step of flattening the images into data vectors to fit the algorithms. For text classification, the preprocessing steps described in Sect. 4.3 were applied.

4.8.1 ViT-Base-Patch16-224

The Vision Transformer (ViT), developed by Google, is a transformer-based model for image classification that processes images as sequences of patches using self-attention. We fine-tuned a pre-trained ViT model on our datasets to adapt it for our tasks. The details of the model are elaborated in [34].

4.8.2 DiT-Base-Finetuned-RVLCDIP

Microsoft's DiT-Base model, initially fine-tuned on the RVLCDIP dataset for document image classification, served as the foundation for our work. We further fine-tuned this model to adapt it to our specific tasks, ensuring that it aligns more closely with the unique requirements and characteristics of our datasets. The model architecture and training details can be found in [35].

4.8.3 XLNet

XLNet is an autoregressive transformer model that leverages a generalized permutation-based training objective to capture bidirectional context. The model architecture and training methodology are described in [36]. In this study, XLNet was used for text classification tasks. The model utilizes a pre-trained XLNet model, which is initialized using the `from_pretrained` function.

4.8.4 BERT

As the name implies, BERT, or Bidirectional Encoder Representations from Transformers, is a transformer architecture model that considers both left and right representations of context. This means that it learns word patterns by combining words in both directions. Several studies have been conducted on this robust architecture in recent years. For instance, [37] demonstrated the model's effectiveness in text classification.

4.9 Evaluation Metrics

After introducing the models used for benchmarking, it is essential to evaluate them based both on response time and classification performance. This section discusses the evaluation metrics used to assess the performance of our model. These metrics provide insight into various aspects of the model's time inference, precision, recall, and F1-score [29].

4.9.1 Time Inference

Time inference measures the computational efficiency of the model by evaluating the time it takes to make predictions on new data. This metric is crucial for understanding the practicality of the model in real-world applications where processing speed may be a critical factor [29]. Time inference is calculated by recording the duration from the moment the model starts making a prediction to the moment it outputs the result. This can be expressed in seconds (s) or milliseconds (ms), depending on the scale of the measured time. To measure the model inference time accurately, we used the `time.process_time()` function from Python's `time` module.

4.9.2 Precision

The precision metric quantifies the model's capacity to predict positive outcomes [29]. This measure, which is defined as the ratio of true positive predictions to the total of true positive and false positive predictions, serves in addressing the question, "Out of all the predictions we made, how many were true?" The following offers mathematical accuracy:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

The formula's true positives (TP) indicate that the actual value was, in fact, positive and that it was anticipated to be such; false positives (FP) indicate that the actual value was negative but that it was predicted to be positive.

4.9.3 Recall

Recall, also known as sensitivity or true positive rate [29], measures the model's ability to identify all relevant instances in a dataset and provides an answer to the question, "Of all the data points that should be predicted as true, how many did we correctly predict as true?" It is defined as the ratio of true positive predictions to the sum of true positives and false negatives. The recall formula is as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

where the number of false negatives (FN) indicates that the actual value was positive when it was predicted to be negative. Since recall measures the model's capacity to catch all real positive cases, it is crucial when the cost of false negatives is high.

4.9.4 F1-Score

The F1-score is a statistic that seeks an equilibrium between precision and recall, calculated as the harmonic mean of the two. When working with unbalanced datasets, when a single statistic might not give a comprehensive view, it is especially helpful. The following formula is used to determine the F1-score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

This metric has a range of 0 to 1, where a value of 1 denotes perfect recall and precision, and a value of 0 denotes poor performance in both metrics. When comparing models where precision and recall are crucial, the F1-score is useful. The F1-score uses the harmonic mean of precision and recall to balance these metrics, avoiding the problems of the arithmetic mean.

5 Experiments and Results

With the evaluation metrics defined, we evaluate the performance of the classifiers after their training. Each classifier is evaluated based on precision, recall, F1-score, and prediction runtime. In addition, random sampling tests are used to compare performance differences between classifiers.

5.1 General Performance of the Classifiers

We have evaluated the performance of machine learning (ML), deep learning (DL), and transformer classifiers based on inference time, precision, recall, and F1-score. For each dataset in Table 1 we generate a performance comparison.

5.1.1 Text Based Classifiers

In the following two Tables 2 and 3, we present the performance results of the 12 classifiers. As observed, the ML models had the fastest prediction times on the CPU, likely due to their simpler architectures compared to DL or transformer-based models. In particular, both the TextCNN and CharCNN models exhibited the lowest overall performance. Therefore, this can be attributed to the inherent nature of DL models, as demonstrated in an additional experiment we conducted but did not include in this analysis. In that experiment, we trained both models on an expanded dataset of approximately 70,000 instances, resulting in an improved accuracy of 90%. However, this suggests that these models may require a larger amount of data to achieve

Table 2 Classifier performance comparison on Cv_text. This table shows the performance of the classifiers trained and evaluated on the Cv_text dataset

Classifier	Time (s)	Precision	Recall	F1-score
SGDClassifier	0.0002	0.99	0.99	0.99
LogisticRegression	0.0002	0.99	0.99	0.99
TextCNN	0.027	0.77	0.57	0.47
CharCNN	0.237	0.82	0.79	0.78
XLNet	0.538	0.99	0.99	0.99
BERT	0.372	0.99	0.99	0.99

Table 3 Classifier performance comparison on Paper_text. This table shows the performance of the classifiers trained and evaluated on the Paper_text dataset

Classifier	Time (s)	Precision	Recall	F1-score
SGDClassifier	0.0007	0.99	0.99	0.99
LogisticRegression	0.0007	0.99	0.99	0.99
TextCNN	0.030	0.84	0.76	0.74
CharCNN	0.237	0.75	0.58	0.51
XLNet	0.460	0.99	0.99	0.99
BERT	0.421	0.99	0.99	0.99

satisfactory results. It is also important to note that the training times for DL models are significantly longer than those for ML models, which completed training in just 10–20 s on an Apple M3 CPU. For transformer models, the estimated training time on an M3 CPU was around 9 h for 5 epochs. However, we did not train these models on a CPU; instead, we used a T4 GPU for the actual training, which took approximately 45–60 min to complete 5 epochs.

5.1.2 Image Based Classifiers

Regarding image-based classification, the following two Tables 4 and 5 present a summary of the performance of the 12 image-based classifiers. Similarly to the text-based classifiers in both tasks (CVs and papers) using only a CPU (Apple M3), the ML classifiers had the shortest inference times. The classifiers with the longest inference times were ConvNextV2, ViT-Base-Patch16, and Dit-base-finetuned-rvlcdip. However, despite their faster inference times, the ML classifiers showed lower image classification performance. This may be due to the complexity of images and the inability of these classifiers to capture spatial patterns as effectively as the DL and transformer architectures. In terms of training times, the ML classifiers required around 2–3 min to train and generally achieved approximately 90% accuracy.

Table 4 Classifier performance comparison on Cv_image. This table shows the performance of the classifiers trained and evaluated on the Cv_image dataset

Classifier	Time (s)	Precision	Recall	F1-score
SGDClassifier	0.001	0.83	0.74	0.72
LogisticRegression	0.001	0.98	0.98	0.98
Convnextv2	0.900	0.99	0.99	0.99
Resnet-50	0.239	0.98	0.98	0.98
ViT-Base-Patch16	0.710	0.91	0.91	0.91
Dit-base-finetuned-rvlcdip	0.559	0.99	0.99	0.99

Table 5 Classifier performance comparison on Paper_image. This table shows the performance of the classifiers trained and evaluated on the Paper_image dataset

Classifier	Time (s)	Precision	Recall	F1-score
SGDClassifier	0.001	0.72	0.51	0.35
LogisticRegression	0.001	0.80	0.67	0.64
Convnextv2	0.595	0.92	0.91	0.91
Resnet-50	0.228	0.99	0.99	0.99
ViT-Base-Patch16	0.694	0.96	0.96	0.96
Dit-base-finetuned-rvlcdip	0.577	0.99	0.99	0.99

In contrast, deep learning and transformer classifiers took approximately 30–45 min to complete 5 epochs of training on the same T4 GPU, and delivered superior results, with an accuracy of around 99%.

5.2 Random Sampling Test

In order to run a final test, we gathered 250 new samples in three categories: scientific articles, non-scientific materials unrelated to CVs, and CVs. Both image and text models were applied to these data in order to evaluate the generalization capacity of the models in dynamic contexts by evaluating their performance on new data and comparing the results with other models. We generated four datasets, each with 500 samples: 250 valid documents and 250 invalid documents, randomly divided into 5 batches of 50 valid documents and 50 invalid documents.

5.2.1 Random Sampling Test on Text Models

After evaluating the models on five randomly generated batches of text data, we found that most of the models maintained their performance. As shown in Figs. 6 and 7, most of the models generalize well to new data in both resumes and scientific articles. This consistency may be due to the relatively stable nature of the text produced in these types of documents. Although the visual format may differ, the grammatical structures and writing styles tend to remain similar over time.

5.2.2 Random Sampling Test on Image Models

For image-based models, we observed that changes in the format of CVs or papers can significantly impact their ability to generalize to new instances. This suggests that choosing an image-based approach may not be ideal in environments where the formatting varies widely. In particular, ML models experienced the most severe drop in performance. In contrast, deep learning and transformer models, although they also saw a reduction in performance, still maintained relatively strong performance despite these changes. This resilience could be attributed to the flexibility of the data used during their pre-training (Figs. 8 and 9).

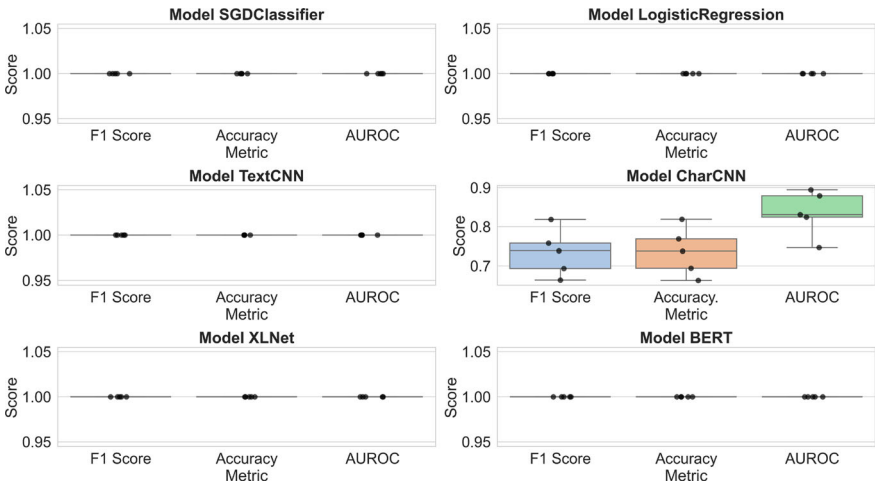


Fig. 6 Random sampling test for Cv_text classifiers. This figure shows the comparison of the performance of classifiers trained on Cv_text evaluated with new data

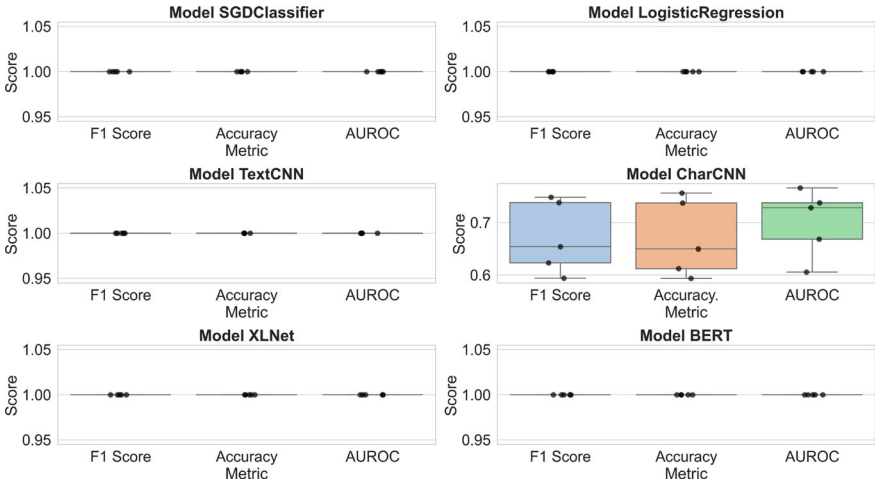


Fig. 7 Random sampling test for Paper_text classifiers. This figure shows the comparison of the performance of classifiers trained on Paper_text evaluated with new data

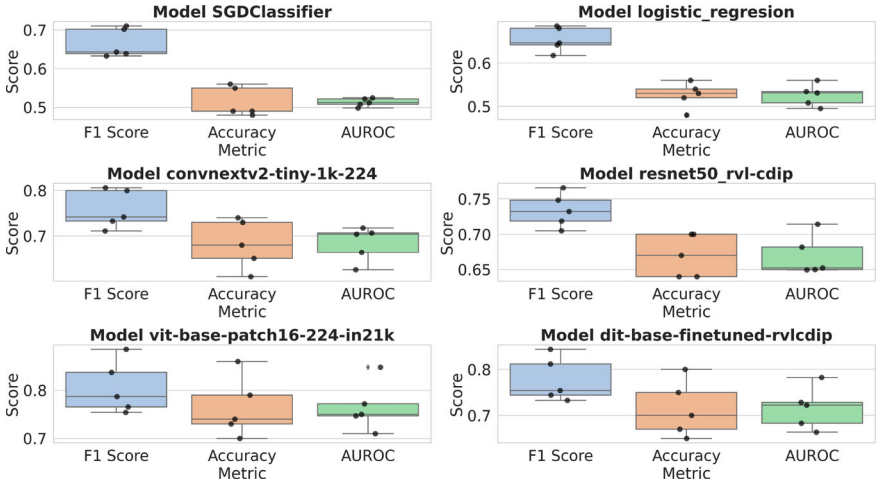


Fig. 8 Random sampling test for Cv_image classifiers. This figure shows the comparison of the performance of classifiers trained on Cv_image evaluated with new data

5.3 Comparative Results with the State of the Art

After conducting the experiments and observing the results, we will compare our findings with the current state of the art. We will discuss and analyze these results to highlight the significance of our contributions.

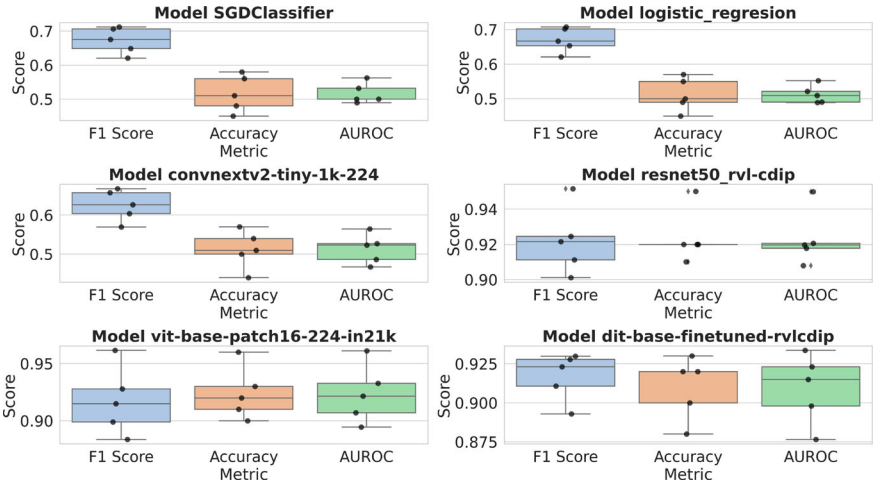


Fig. 9 Random sampling test for Paper_image classifiers. This figure shows the comparison of the performance of classifiers trained on Paper_image evaluated with new data

Table 6 Performance comparison of classifiers across different tasks. This table shows the performance of trained and evaluated classifiers on different tasks, where I compare my machine learning classifiers from my work with a spam filter classifier from another study

Classifier	Task	Time (s)	Precision	Recall	F1-score
SGDClassifier	Cv detection	0.0002	0.99	0.99	0.99
LogisticRegression	Cv detection	0.0002	0.99	0.99	0.99
SGDClassifier	Paper detection	0.0007	0.99	0.99	0.99
LogisticRegression	Paper detection	0.0007	0.99	0.99	0.99
Random Forest	Spam detection	0.025	0.94	0.94	0.94
XGBoost	Spam detection	0.017	0.94	0.93	0.94

Although there are works that address document classification [35, 38], they do not address the problem in the same way as our work using ML, deep learning, and transformers that use images or text as input.

For the Table 6, we choose the two best classifiers from the work [29] and compare them with our text-based ML classifiers. Our ML models deliver 46 times faster inference and a 0.05 improvement in performance. Although they are designed for different tasks, incorporating techniques such as hyperparameter optimization, TF-IDF, or embeddings could further enhance the performance of spam detection classifiers.

5.3.1 Discussion and Analysis of Results

Depending on the resources at your disposal and the type of data you are working with (whether text or images), certain classifiers may be more suitable than others. For text data on a CPU, traditional machine learning classifiers are the best choice. If you have a GPU and are working with text data, transformer-based classifiers are optimal and offer superior performance. For image data on a CPU, traditional machine learning classifiers train faster but may not achieve the best performance. However, if you have a GPU and are working with image data, transformer-based classifiers provide the best performance. In general, classifiers trained on text data outperform those trained on image data.

6 Conclusion and Future Work

This work demonstrates that deep learning is not always the optimal solution for all classification tasks, particularly when data constraints or computational efficiency are critical. We benchmarked traditional machine learning (ML), deep learning, and transformer models for document classification using both text and image data, offering several insights and contributions. Our first contribution is the design of two integrated pipelines: one for text extraction and another for the training and evaluation of 24 classifiers, 12 for each task, on four datasets comprising both internal and open-source data. Secondly, we found that traditional ML models often outperform deep learning for text classification tasks due to their faster inference times and competitive accuracy, especially in scenarios with limited training data. In contrast, transformer-based models maintain strong performance across tasks, while deep learning models struggle without sufficient data. Thirdly, our results highlight the effectiveness of fine-tuning pre-trained deep learning and transformer models for image classification, where they significantly outperform traditional ML approaches. This demonstrates the advantage of leveraging pre-trained knowledge for visual tasks. We also observed that text classifiers generalize well to new data, thanks to the stable vocabulary used in CVs, whereas image classifiers face challenges due to the diversity of image formats and styles. Finally, our work provides practical insights into the trade-offs between computational efficiency and accuracy. While traditional ML models are suitable for fast inference with text data, deep learning and transformer models, despite their computational demands, are better suited for complex tasks like image classification.

In future work, we plan to explore a hybrid approach that integrates both text and image inputs rather than handling them independently. By combining these modalities, we aim to leverage the complementary information that text and images can provide when processed together. This approach may enhance the model's ability to understand and generate more contextually relevant output, leading to more accurate and robust results. Additionally, we will investigate the potential use of

quantum computing to optimize feature extraction and similarity matching. Quantum-enhanced embeddings and hybrid quantum-classical models could improve processing efficiency, especially for high-dimensional data, advancing both performance and scalability.

Acknowledgements This research was made possible by the ITJUANA, S. DE R.L. DE C.V. company through its ITJ Labs department. Furthermore, we extend our gratitude to the Instituto Politécnico Nacional for the invitation to the Seminar AI&QC. We are truly grateful for the contributions and support of the ITJ Labs team, which has played a crucial role in successfully completing this endeavor.

References

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
2. Tunstall, L., Werra, L.: Natural Language Processing with Transformers: Building Language Applications with Hugging Face. O'Reilly Media, Sebastopol, CA, USA (2023)
3. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems **36** (2024)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Moutik, O., Sekkat, H., Tigani, S., Chehri, A., Saadane, R., Tchakoucht, T.A., Paul, A.: Convolutional neural networks or vision transformers: Who will win the race for action recognitions in visual data? Sensors **23**(2), 734 (2023)
6. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 991–995 (2015). IEEE
7. Géron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Sebastopol, CA, USA (2019)
8. Hecht-Nielsen, R.: In: Wechsler, H. (ed.) III.3 - Theory of the Backpropagation Neural Network, pp. 65–93. Academic Press, San Diego, CA, USA (1992). https://doi.org/10.1016/B978-0-12-741252-8.50010-8 . https://www.sciencedirect.com/science/article/pii/B9780127412528500108
9. Sanders, J., Kandrot, E.: CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, Upper Saddle River, NJ, USA (2010)
10. Pradhan, S.O., Susarla, D.: Principles of Data Science - Third Edition: A Beginner's Guide to Essential Math and Coding Skills for Data Fluency and Machine Learning, 3rd edn. Packt Publishing, Birmingham, UK (2022)
11. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer, New York (2010)
12. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 665–666 (2006)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

14. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229 (2020). Springer
15. IBM: Natural Language Processing. Accessed: 2024-08-06 (2024). https://www.ibm.com/topics/natural-language-processing
16. Yamamoto, M.: Evolution of natural language processing technology: Not just language processing towards general purpose ai. arXiv preprint arXiv:2310.06228 (2023)
17. Halevy, A., Norvig, P., Pereira, F.: The unreasonable effectiveness of data. IEEE intelligent systems **24**(2), 8–12 (2009)
18. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 843–852 (2017)
19. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2016). IEEE
20. Wang, P., Fan, E., Wang, P.: Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. Pattern recognition letters **141**, 61–67 (2021)
21. Bakkali, S., Ming, Z., Coustaty, M., Rusiñol, M.: Eaml: ensemble self-attention-based mutual learning network for document image classification. International Journal on Document Analysis and Recognition (IJDAR) **24**(3), 251–268 (2021)
22. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 993–1003 (2021)
23. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: Layoutlmv3: Pre-training for document ai with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4083–4091 (2022)
24. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. IEEE transactions on neural networks and learning systems **32**(2), 604–624 (2020)
25. González-Carvajal, S., Garrido-Merchán, E.C.: Comparing bert against traditional machine learning text classification. arXiv preprint arXiv:2005.13012 (2020)
26. Nair, B.J.B., Yadhukrishnan, S., Manish, A.: A comparative study on document images classification using logistic regression and multiple linear regressions. Proceedings of the 2023 2nd International Conference on Augmented Intelligence and Sustainable Systems, ICAISS 2023, 1096–1104 (2023) https://doi.org/10.1109/ICAISS58487.2023.10250671
27. Zingaro, S.P., Lisanti, G., Gabbrielli, M.: Multimodal side-tuning for document classification. Proceedings - International Conference on Pattern Recognition, 5206–5213 (2023) https://doi.org/10.1109/ICPR48806.2021.9413208
28. Jiechieu, K.F.F., Tsopze, N.: Skills prediction based on multi-label resume classification using cnn with model predictions explanation. Neural Computing and Applications **33**, 5069–5087 (2021) https://doi.org/10.1007/s00521-020-05302-x
29. Occhipinti, A., Rogers, L., Angione, C.: A pipeline and comparative study of 12 machine learning models for text classification. Expert Systems with Applications **201**, 117193 (2022) https://doi.org/10.1016/J.ESWA.2022.117193
30. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders (2023). https://arxiv.org/abs/2301.00808
31. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference (2014). https://doi.org/10.3115/v1/d14-1181
32. Gong, L., Ji, R.: What Does a TextCNN Learn? (2018). https://arxiv.org/abs/1801.06287
33. Zhang, X., Zhao, J., Lecun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, vol. 2015-January (2015)

34. Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., Vajda, P.: Visual Transformers: Token-based Image Representation and Processing for Computer Vision (2020). https://arxiv.org/abs/2006.03677
35. Lewis, D., Lewis, D.D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a Test Collection for Complex Document Information Processing. http://legacy.library.ucsf.edu/legal.html
36. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
37. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1 (2019)
38. Wagh, V., Khandve, S., Joshi, I., Wani, A., Kale, G., Joshi, R.: Comparative study of long document classification. In: TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), pp. 732–737. IEEE, Auckland, New Zealand (2021). https://doi.org/10.1109/TENCON54134.2021.9707465

Artificial Intelligence in Astrophysics: Analytical Enhancements

Luis H. Sanchez, Carlos A. Guerrero, Juan J. Tapia, and Ana V. Ojeda

Abstract The study of the cosmos through large-scale surveys such as J-PLUS and Gaia generates vast amounts of astronomical data, ushering astronomy into the era of big data. These observations span a wide range of energies, from gamma rays to radio waves, producing petabytes of data that require advanced techniques for efficient processing and analysis. The application of artificial intelligence (AI) and machine learning, particularly in the estimation of astrophysical parameters like stellar temperature, luminosity, and radius, has become crucial in this context. Models such as XGBoost and Random Forest are increasingly used to predict these parameters, aiding in the understanding of stellar evolution and galactic dynamics. However, the success of these models is contingent on the quality of the data, making preprocessing a critical step. This chapter focuses on data cleaning, correction of interstellar extinction, and feature engineering to enhance the accuracy of machine learning predictions. By integrating AI, this work demonstrates how modern techniques can transform our understanding of the universe.

Keywords Astrophysics · Artificial intelligence · Machine learning · Feature engineering · Photometric data

L. H. Sanchez (✉) · J. J. Tapia · A. V. Ojeda
Instituto Politécnico Nacional, Tijuana, Mexico
e-mail: lsanchez@citedi.mx

J. J. Tapia
e-mail: jtapiaa@ipn.mx

A. V. Ojeda
e-mail: aojeda@citedi.mx

C. A. Guerrero
Universidad Autónoma de Nuevo León, Monterrey, Mexico
e-mail: carlos.guerrerop@uanl.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_19

1 Introduction

The study of the universe through observations from Earth and space has led to the massive collection of astronomical data, covering a wide spectrum from gamma rays to radio waves and other cosmic messengers such as neutrinos and cosmic rays [1]. Instruments like J-PLUS (Javalambre Photometric Local Universe Survey) [2] and the Gaia mission [3] have been responsible for generating large volumes of data, presenting new challenges in terms of storage, processing, and efficient analysis.

J-PLUS is an astronomical mission whose main goal is to observe a significant portion of the sky using 12 passband photometric filters, each designed to capture different ranges of the electromagnetic spectrum [4]. These filters, ranging from ultraviolet to near-infrared, allow astronomers to analyze the light emitted by astronomical objects at different wavelengths, providing crucial information about their physical properties. The use of these filters is fundamental for creating color indices, which are key to inferring astrophysical parameters such as temperature and stellar composition [5].

A critical aspect of stellar analysis is the relationship between temperature, radius, and luminosity of a star, described by Stefan-Boltzman's Law, see Eq. 1:

$$L = 4\pi\sigma r^2 T^4, \tag{1}$$

where L is the luminosity, r the radius, T the effective temperature, and σ the Stefan-Boltzmann constant [6]. This law is crucial for understanding how the color indices, obtained through J-PLUS filters, can be used to estimate astrophysical parameters such as temperature and luminosity.

The vast amount of data generated by J-PLUS, combined with the detailed astrometric information provided by Gaia, has opened the door to new methods for estimating astrophysical parameters. These methods leverage artificial intelligence (AI) techniques, particularly machine learning models, to make predictions based on large datasets. The quality of these predictions depends heavily on the preprocessing of the data, which includes steps such as outlier removal, correction of interstellar extinction, and data correlation between different catalogs.

This chapter explores the application of AI in the analysis of astronomical data, focusing on preprocessing techniques and the estimation of astrophysical parameters such as effective temperature, luminosity, and stellar radius. In addition, it covers the evaluation methods and techniques used to ensure the accuracy of predictions, emphasizing the mathematical foundations behind these applications.

2 Data and Preprocessing

The datasets from the J-PLUS DR3 and Gaia DR3 catalogs provide rich photometric and astrometric data, allowing for the study of millions of celestial objects. However, these data require extensive preprocessing to ensure their quality and suitability for

machine learning models. This section presents the detailed workflow for preparing the data, including catalog matching, adjustments for interstellar extinction, and identifying and removing outliers. These preprocessing steps play a pivotal role in refining the dataset to enable precise estimations of astrophysical parameters such as effective temperature, luminosity, and stellar radius.

2.1 Data Sources

2.1.1 J-PLUS DR3 Photometric Data

The J-PLUS DR3 catalog contains photometric data for more than 43 million celestial objects, collected using a system of 12 passband filters that cover different regions of the electromagnetic spectrum, from ultraviolet to near-infrared [2]. These filters allow astronomers to capture detailed information about the light emitted by celestial objects at different wavelengths, which is critical for inferring physical properties through the analysis of the spectral energy distribution.

2.1.2 Gaia DR3 Astrometric Data

In addition to the photometric data from J-PLUS, the Gaia DR3 catalog provides astrometric data, which includes precise measurements of celestial object positions, parallaxes, and proper motions for over 1.8 billion stars [3]. By combining these astrometric properties with the photometric data from J-PLUS, a more complete dataset is created, providing essential data for determining astrophysical properties, including distances, stellar dimensions, and brightness levels.

2.2 Data Correlation Between J-PLUS and Gaia

The correlation of data between J-PLUS and Gaia DR3 is an essential step to enrich the dataset, incorporating both photometric and astrometric information to better estimate astrophysical parameters. A total of 23,699,393 records were successfully correlated between these two catalogs [7]. This process was conducted by matching celestial objects from both catalogs through shared identifiers, resulting in a unified dataset containing photometric data from J-PLUS and astrometric data from Gaia.

This correlation enriched the dataset by incorporating astrometric properties like parallax and proper motion, enhancing our capacity to calculate spatial positioning and intrinsic luminosity of celestial objects.

2.3 *Exploratory Data Analysis and Data Cleaning*

Exploratory Data Analysis (EDA) was performed to gain insights into the statistical properties and distributions of the photometric data, laying the groundwork for further preprocessing and model building. This analysis included the computation of key statistics such as mean, median, standard deviation, skewness, and asymmetry across the 12 J-PLUS passbands. Visualizations like Color-Magnitude Diagrams (CMDs) and Color-Color Diagrams (CCDs) were used to explore relationships between magnitudes, color indices (comparison between two or more passbands) and stellar parameters [8, 9].

Additionally, a correlation analysis was performed to investigate the relationships between J-PLUS photometric features and astrophysical parameters from Gaia DR3, such as effective temperature and luminosity. The correlation matrix (Fig. 1) revealed a strong relationship between the various photometric bands themselves, reflecting

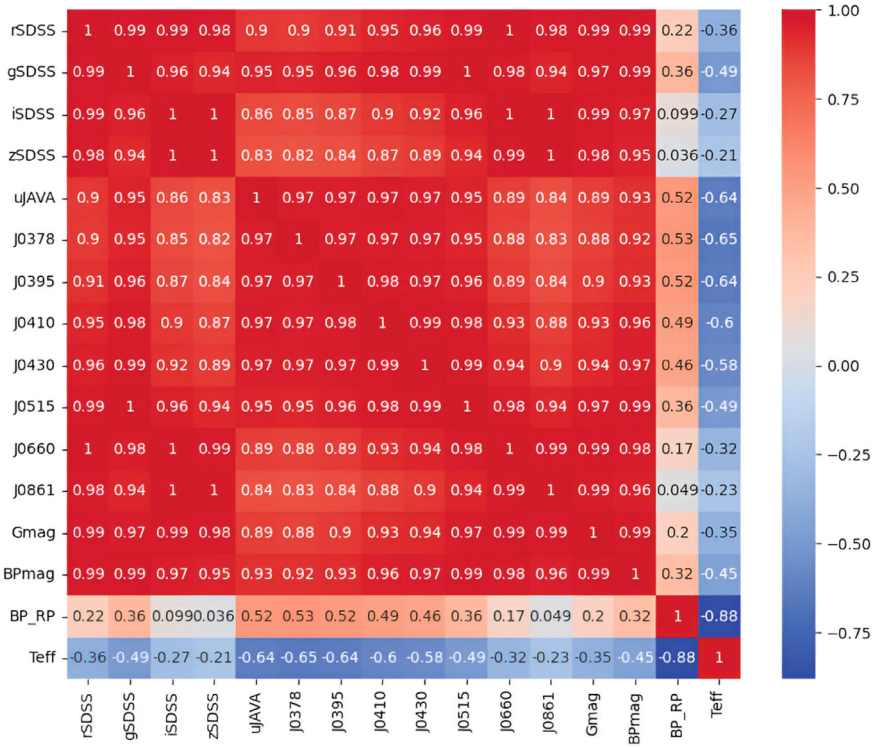


Fig. 1 Correlation matrix among the photometric bands of J-PLUS and stellar effective temperature (T_{eff}) from Gaia DR3

the inherent redundancy in multi-filter photometric surveys [8]. The combination of multiple features will be crucial for improving machine learning models aimed at parameter estimation.

2.4 Reddening Correction and Interstellar Extinction

One of the most significant challenges in working with photometric data is the distortion caused by interstellar dust, which leads to reddening and extinction. This effect causes stars to appear redder and fainter than they actually are, resulting in potential misestimation of their temperature and luminosity. Therefore, correcting for interstellar reddening is a crucial preprocessing step.

To perform this correction, the methodology outlined by López-Sanjuan et al. [10] was applied, using stellar and white dwarf loci to calibrate large-area multi-filter surveys. The correction is expressed by the Eq. 2:

$$X_0 = X - K_x \cdot E(B - V), \tag{2}$$

where X_0 is the corrected magnitude, X is the observed magnitude, K_x is the extinction coefficient specific to each filter, and $E(B - V)$ is the color excess representing the effect of interstellar extinction [11]. This correction ensures that the photometric magnitudes accurately reflect the intrinsic properties of the stars by accounting for the influence of interstellar dust.

2.5 Outlier Removal and Data Preparation

During the EDA, it was observed that measurements with magnitudes greater than 23 exhibited increased noise, potentially distorting model predictions [12]. To mitigate this, a magnitude cutoff of 23 was applied, ensuring the inclusion of data points with higher signal-to-noise ratios.

Outliers were further identified using box plots across the different photometric filters, as shown in Fig. 2. These box plots highlight extreme values for each filter, providing a visual confirmation of data points that deviate significantly from the majority. Based on this analysis, outliers were removed to enhance data consistency and improve the reliability of model training [8].

Following the application of these cleaning steps, the dataset was prepared for feature selection and machine learning model development. This preprocessing ensures a robust and high-quality dataset for estimating parameters such as effective temperature, luminosity, and stellar radius.

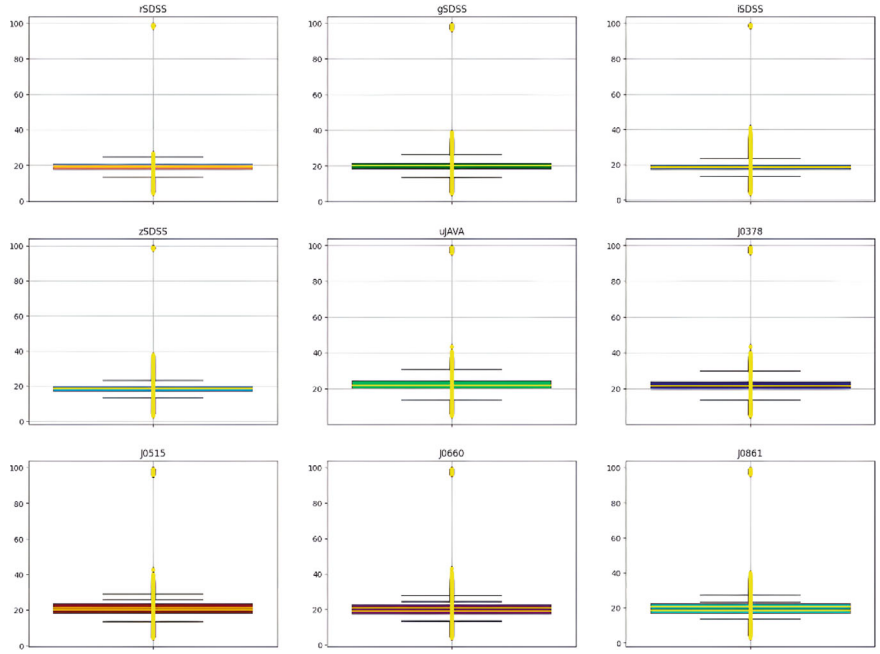


Fig. 2 Box plots illustrating outliers across different photometric filters. Extreme values were identified and removed to improve data quality

2.6 Creation of Color Indices

Astronomical color indices are fundamental for estimating parameters such as stellar temperature and luminosity. These indices are calculated by subtracting the magnitudes of two photometric bands, such as $g - r$ (blue minus visual) or $r - i$ (visual minus infrared) [13].

In this work, color indices were computed using the J-PLUS photometric bands, focusing on combinations that are most sensitive to variations in temperature and metallicity in stars. The indices $g - r$ and $r - i$ (see Eq. 3), were particularly effective in capturing temperature gradients, which are directly related to the effective temperature (T_{eff}) and luminosity of the stars, essential parameters for understanding stellar evolution.

$$\text{Color}_1 = g - r, \quad \text{Color}_2 = r - i, \tag{3}$$

where g, r, and i represent magnitudes in different bands.

Figure 3 presents the Hertzsprung-Russell (H-R) diagram from Gaia DR3, which visually demonstrates the relationship between color indices (e.g., $G_{\text{BP}} - G_{\text{RP}}$) and stellar luminosity. The H-R diagram is a key tool in astronomy, plotting stars based on their luminosity and surface temperature to reveal distinct evolutionary groups

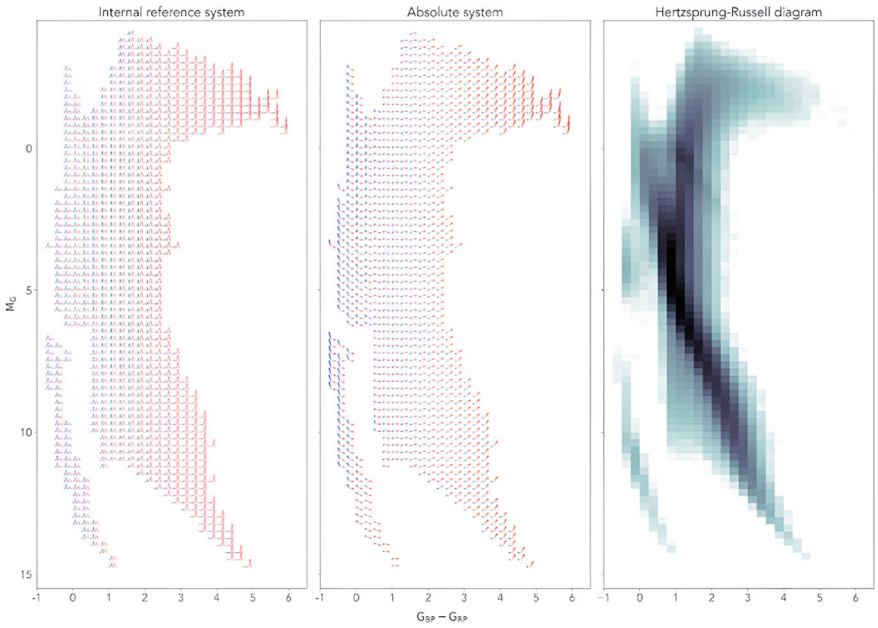


Fig. 3 Hertzsprung-Russell (H-R) diagram from Gaia DR3, illustrating the distribution of stars according to color and luminosity [14]

such as the main sequence, giants, and white dwarfs. This diagram provides insights into stellar characteristics and life cycles, serving as a statistical reference for understanding the distribution of stars according to color and luminosity, and supporting the role of color indices in parameter estimation.

2.7 Estimation of Effective Temperature with XGBoost

The effective temperature (T_{eff}) of a star is one of the key parameters used to characterize its physical properties and evolutionary state. For the estimation of effective temperature, the XGBoost algorithm was employed [15], which is a gradient-boosting method based on decision trees that iteratively minimizes the loss function.

2.7.1 Mathematical Formulation

The estimation of effective temperature is modeled as a regression problem. The goal is to find a function $f : \mathbb{R}^n \to \mathbb{R}$ that maps the astronomical features $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ to the effective temperature T_{eff}. This is achieved by minimizing the mean squared error (MSE) between the true values and the predictions:

$$L(T_{\text{eff}}, f(\mathbf{x})) = \frac{1}{m} \sum_{i=1}^{m} \left(T_{\text{eff},i} - f(\mathbf{x}_i)\right)^2, \tag{4}$$

where m is the number of training examples, $T_{\text{eff},i}$ is the true effective temperature for the i-th star, and $f(\mathbf{x}_i)$ is the model's prediction.

The XGBoost model optimizes this loss function iteratively, and the prediction for the temperature can be written as a sum of several decision trees:

$$f(\mathbf{x}) = \sum_{t=1}^{T} f_t(\mathbf{x}), \tag{5}$$

where $f_t(\mathbf{x})$ is the prediction of the t-th tree, and T is the total number of trees. XGBoost uses L_1 and L_2 regularization to penalize model complexity, preventing overfitting and improving the generalization capacity of the model [15].

2.8 *Principal Component Analysis for Dimensionality Reduction*

Before training the XGBoost model, Principal Component Analysis (PCA) was applied to the dataset to reduce the dimensionality and address multicollinearity issues identified during the exploratory data analysis. PCA transforms the original set of features into a new set of uncorrelated variables, known as principal components, which capture the most variance in the data.

2.8.1 Mathematical Foundations of PCA

Given an input matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m represents the number of stars and n the number of features (e.g., photometric magnitudes, morphological properties, parallaxes), PCA computes the covariance matrix of the data:

$$\Sigma = \frac{1}{m-1} \mathbf{X}_c^\top \mathbf{X}_c, \tag{6}$$

where $\mathbf{X}_c$ is the centered data matrix (data minus the mean). The eigenvalues and eigenvectors of Σ are then computed:

$$\Sigma \mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{7}$$

where λ_i are the eigenvalues and $\mathbf{v}_i$ are the corresponding eigenvectors. The eigenvalues represent the variance explained by each principal component, and the eigenvectors define the directions in the feature space along which this variance occurs.

The data is projected onto the principal components as follows:

$$\mathbf{Z} = \mathbf{X}_c \mathbf{V}, \tag{8}$$

where $\mathbf{V}$ is the matrix of eigenvectors, and $\mathbf{Z}$ contains the transformed features (principal components). PCA allows us to retain only the components that explain most of the variance, thereby reducing the number of features while keeping the majority of the information.

2.8.2 Application of PCA in This Study

In this study, PCA was applied to the original features (including photometric magnitudes, morphological properties, and parallax). The analysis revealed that 9 principal components explained 90% of the variance in the data, as shown in Fig. 4. These components were then used as input for the XGBoost model.

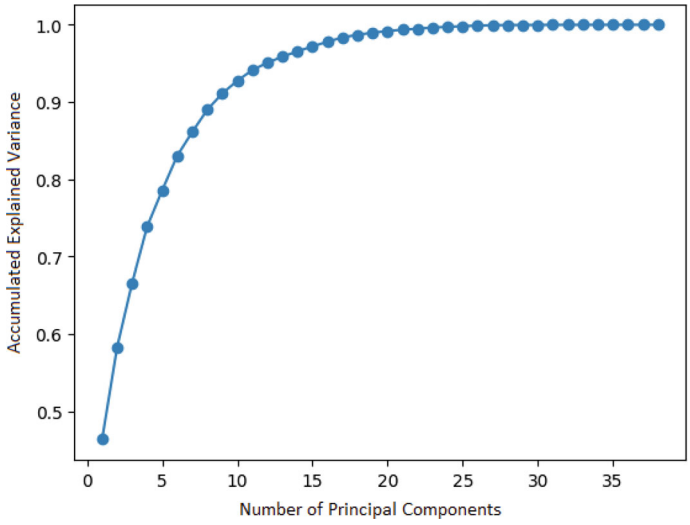


Fig. 4 Accumulated explained variance for principal components

2.8.3 Interpretation of Principal Components

The weights of the original features in the principal components are visualized in Fig. 5. These weights show how each original feature contributes to the principal components. Features with higher absolute values in the loadings have a greater influence on the corresponding principal component. By examining these weights, we can gain insights into which features (e.g., photometric magnitudes, morphological properties, parallax) are most relevant for estimating effective temperature.

The third principal component (PC2) was found to have the greatest influence on the model's predictions for effective temperature (Fig. 6). This component is primarily influenced by photometric and astrometric features, such as *e_RaDeg*, *PM*, and *Gmag*, which may provide indirectly relevant information for estimating effective temperature.

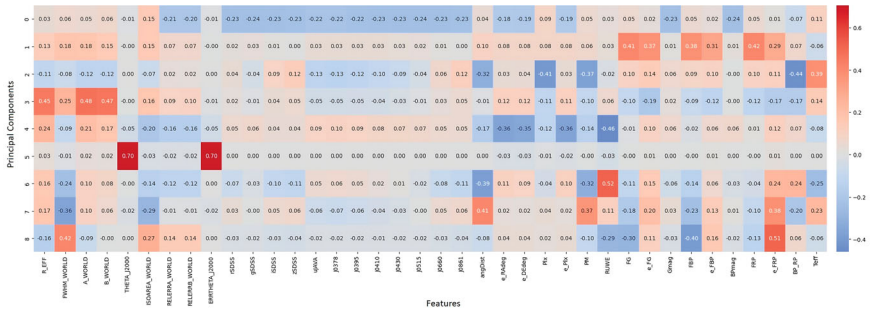


Fig. 5 Feature weights in principal components

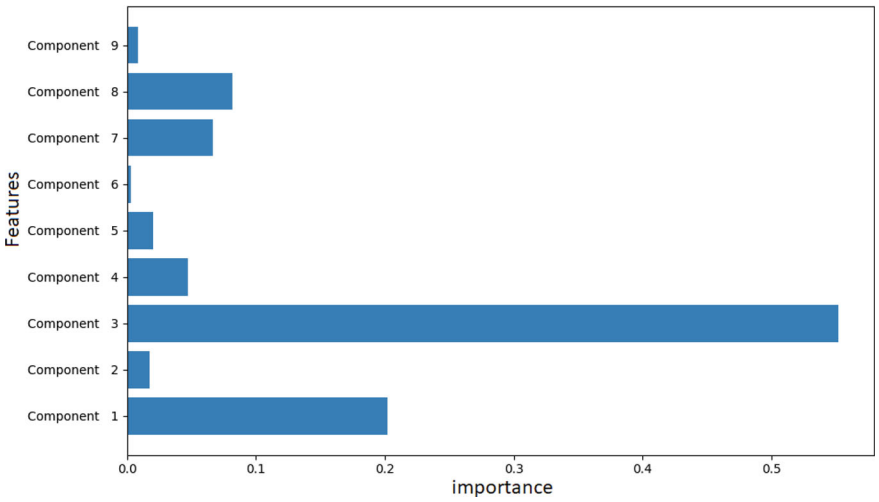


Fig. 6 Feature importance in the XGBoost model

2.9 Estimation of Stellar Radius with XGBoost

The stellar radius (R) is a fundamental parameter for understanding the physical characteristics and evolutionary stages of stars. For the estimation of stellar radius, the XGBoost algorithm was employed, leveraging the photometric and astrometric data available from the J-PLUS and Gaia catalogs.

2.9.1 Mathematical Formulation

The estimation of stellar radius was formulated also as a regression problem. The goal is to find a function $f : \mathbb{R}^n \to \mathbb{R}$ that maps the astronomical features $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ to the stellar radius R. The objective is to minimize the MSE between the true values and the predicted values:

$$L(R, f(\mathbf{x})) = \frac{1}{m} \sum_{i=1}^{m} (R_i - f(\mathbf{x}_i))^2 , \tag{9}$$

where m is the number of training examples, R_i is the true stellar radius for the i-th star, and $f(\mathbf{x}_i)$ is the model's prediction (see Sect. 2.7.1).

2.9.2 Data and Feature Selection

The dataset used for the estimation of stellar radius was constructed from the J-PLUS DR3 survey, including stars with magnitudes below 23 to ensure high-quality data. The astrometric data from Gaia DR2, including parallax, proper motion, and derived features such as luminosity and effective temperature, were used as key inputs for the model. A total of 1,771,578 stars with a *class_star* rating of 0.9 or higher were included in the analysis. The *class_star* parameter, originally introduced in the source extraction software SExtractor, is an indicator of stellarity that quantifies the likelihood of an object being a star rather than an extended source such as a galaxy [16]. This threshold was chosen to ensure that only objects with a high probability of being stars were included, reducing contamination from non-stellar sources. This approach aligns with methods used in large astronomical surveys, such as Gaia, where parameters related to source classification are essential to maintain data quality [3].

Key features used for training included:

- Photometric magnitudes from the 12 J-PLUS passbands,
- Astrometric data from Gaia DR2, such as parallax and proper motion,
- Derived parameters, such as effective temperature and luminosity,
- Color indices, such as $G_{BP} - G_{RP}$ from Gaia.

No dimensionality reduction, such as PCA, was applied, as the features selected for this task were already optimized based on the EDA.

2.9.3 Feature Importance and Interpretation

The XGBoost model inherently calculates the importance of each feature based on how much each feature contributes to reducing the loss function across the decision trees. In this case, the most significant contributors to the prediction of stellar radius were luminosity and effective temperature. This aligns with the theoretical relationship described by the Stefan-Boltzmann La (see Eq. 1).

The model's results confirm that luminosity and temperature are critical for predicting stellar radius, reinforcing the physical understanding of stellar properties.

The feature importance calculated by the model is visualized in Fig. 7, which shows the relative contribution of each feature to the prediction of stellar radius. As expected, luminosity and temperature are the most influential features.

2.10 Estimation of Stellar Luminosity with Random Forest

The stellar luminosity (L) is a crucial parameter in understanding the energy output of stars and plays a significant role in characterizing their physical and evolutionary stages. For the estimation of stellar luminosity, a Random Forest model was

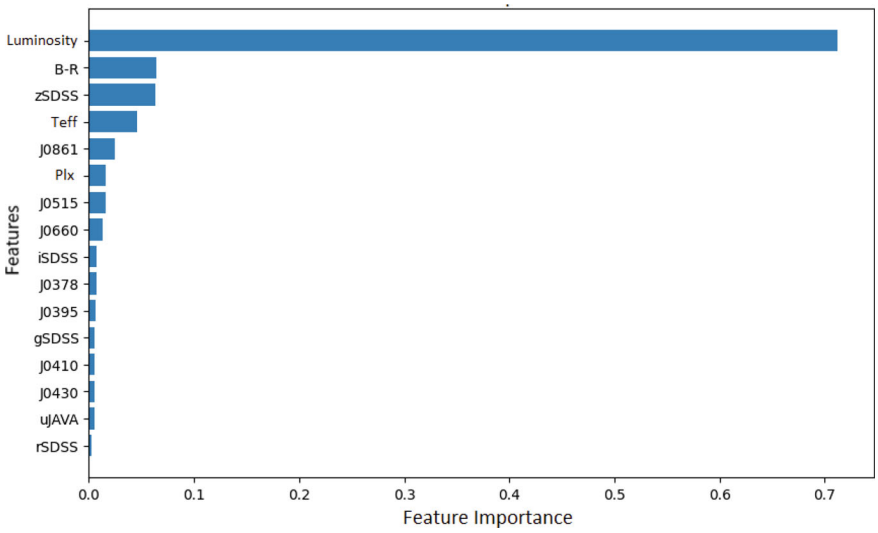


Fig. 7 Feature importance in XGBoost model for stellar radius

employed. This method, unlike XGBoost, was selected to showcase the versatility of ensemble methods in different contexts and the ability of Random Forest to handle variance effectively without requiring intensive hyperparameter tuning, making it particularly suitable for this study's exploratory approach to tool application [17].

2.10.1 Mathematical Formulation

The estimation of stellar luminosity is framed as a regression problem. The Random Forest model consists of an ensemble of decision trees, each trained on a subset of the data. The prediction for luminosity is derived by averaging the outputs of individual trees:

$$f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}), \tag{10}$$

where N is the number of trees in the forest, and $f_i(\mathbf{x})$ represents the individual prediction of each tree for the input features $\mathbf{x}$. Averaging these predictions reduces variance and increases the model's robustness against overfitting, making it particularly effective for the luminosity estimation task, where a range of photometric and astrometric features are utilized.

2.10.2 Data and Feature Selection

The dataset used for the estimation of stellar luminosity was obtained from the J-PLUS DR2 catalog. Stars with magnitudes below 23 were selected to ensure data quality, and a *class_star* threshold of 0.9 was applied to focus on objects classified with high confidence as stars. After applying these selection criteria, the dataset included a total of 1,529,936 stellar objects.

The features used in the training process included:

- Photometric magnitudes from the 12 J-PLUS passbands,
- Astrometric data from Gaia DR2, specifically parallax, which is theoretically aligned with distance and thus directly influences luminosity [3],
- Derived color indices, such as $G_{BP} - G_{RP}$ and $g - r$, capturing key stellar characteristics relevant to luminosity.

In contrast to the approach used for temperature estimation, no dimensionality reduction techniques, such as PCA, were applied, as the selected features were found to be informative and manageable in their raw form.

2.10.3 Feature Importance and Interpretation

The Random Forest model provides a measure of feature importance by calculating the contribution of each feature in reducing the prediction error across all trees. According to the feature importance analysis, parallax emerged as the most significant feature in predicting stellar luminosity, aligning with theoretical expectations, given its direct relationship to distance and the inverse-square law that governs luminosity [18]. Color indices, particularly $G_{BP} - G_{RP}$, also contributed to the model but with less impact than parallax, indicating their supplementary role in luminosity estimation.

The importance of each feature for predicting stellar luminosity is visualized in Fig. 8, where parallax is shown as the most influential feature in the model.

3 Results and Discussion

This section presents the performance evaluation of the models for estimating effective temperature, stellar radius, and luminosity, alongside a detailed discussion of the metrics used. Scatter plots illustrate the model accuracy, and residual analysis provides insights into prediction errors.

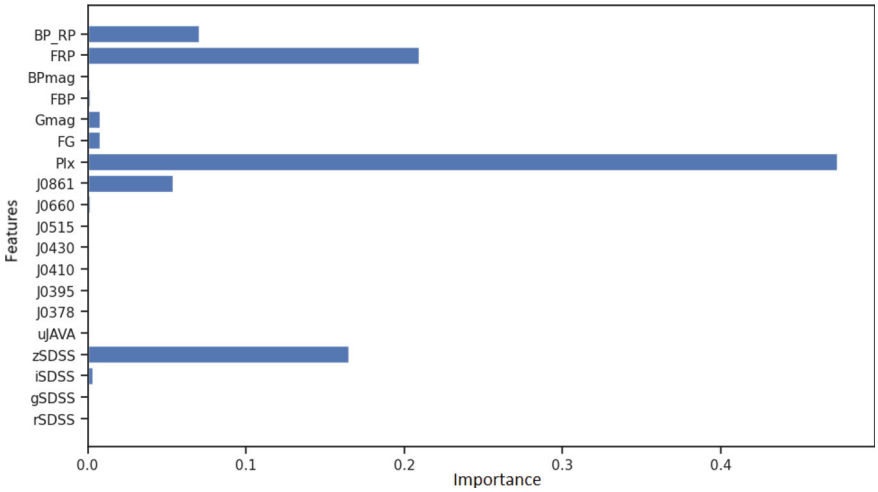


Fig. 8 Feature importance in Random Forest model for stellar luminosity

3.1 Model Evaluation Metrics

To quantitatively assess model accuracy and reliability, we used the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). These metrics help determine how well the models generalize across wide ranges of stellar parameter.

3.1.1 Mean Squared Error

The MSE calculates the average squared difference between the actual values y_i and predicted values $\hat{y}_i$, serving as a measure of the average error magnitude:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2, \tag{11}$$

where m is the number of samples. Lower MSE values indicate a closer fit to the data, while larger values suggest higher prediction error.

3.1.2 Root Mean Squared Error

The RMSE, defined as the square root of MSE, provides error measurements in the same units as the predicted values, enhancing interpretability:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}. \tag{12}$$

RMSE is particularly sensitive to large errors, making it valuable in detecting high variance in predictions.

3.1.3 Mean Absolute Error

MAE represents the average absolute difference between predictions and actual values, making it less sensitive to large outliers:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|. \tag{13}$$

Lower MAE values suggest more accurate predictions, reflecting a model's robustness in capturing the overall trends without being disproportionately influenced by outliers.

3.1.4 Coefficient of Determination

The R^2 value quantifies the proportion of variance explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}, \tag{14}$$

where $\bar{y}$ is the mean of the true values. An R^2 close to 1 indicates that the model explains most of the variability in the data, while values closer to 0 suggest limited explanatory power.

3.2 Effective Temperature Estimation

The XGBoost model achieved an R^2 value of approximately 0.95, indicating that the model effectively captures the non-linear relationships between photometric features and effective temperature. Figure 9 presents a scatter plot of predicted versus actual temperature values, demonstrating close alignment along the diagonal. This alignment indicates strong model accuracy across a wide range of temperatures. Residual analysis revealed minor errors in extreme temperature values, suggesting that incorporating additional astrophysical features could further reduce these discrepancies.

3.3 Stellar Radius Estimation

For stellar radius estimation, the model demonstrated strong predictive accuracy, achieving an R^2 value of 0.98 and low error metrics. Figure 10 illustrates the predicted versus actual radius values, showing a close distribution along the diagonal, which indicates accurate predictions across a range of stellar radii. Residual analysis confirmed that prediction errors were minimal, with some outliers observed for high radius values. This performance reinforces the model's suitability for estimating stellar radius. Future enhancements could be achieved by incorporating more detailed astrophysical features, such as additional spectral indices or refined photometric measurements, which may further improve model accuracy.

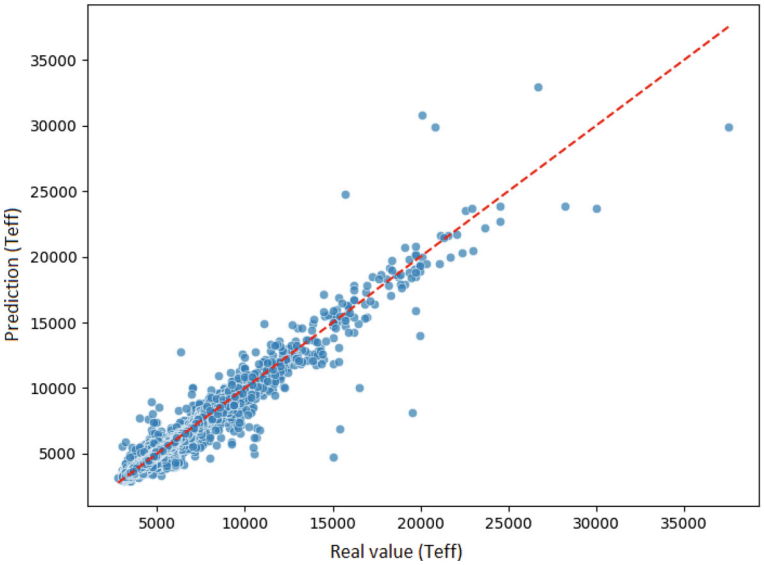


Fig. 9 Scatter plot of effective temperature predictions using XGBoost

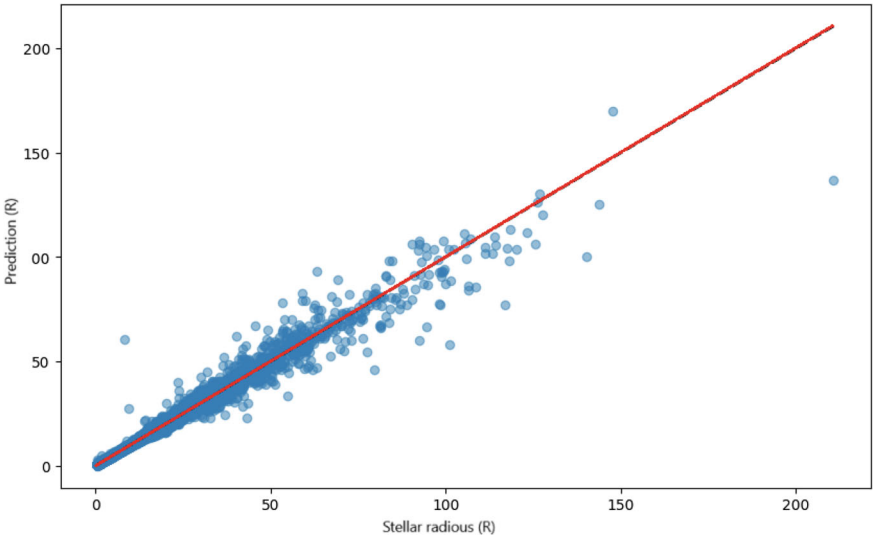


Fig. 10 Scatter plot for stellar radius predictions

3.4 Luminosity Estimation

The Random Forest model, used for luminosity estimation, achieved an R^2 value close to 0.92, validating its ability to capture complex relationships between photometric data and stellar luminosity. The scatter plot in Fig. 11 displays predicted versus actual luminosity values, with strong alignment around the diagonal line. Cross-validation revealed minimal variance in prediction error, confirming the model's robustness across different subsets of data. The reliance on parallax as a primary feature aligns with theoretical expectations, as parallax is directly related to distance, a fundamental factor in luminosity estimation.

3.5 Residual and Error Analysis

Residual analysis for each parameter estimation revealed minor prediction errors for most values but larger residuals in extreme cases, particularly in high temperatures and luminosities. These patterns suggest that additional features, such as metallicity or unique chemical abundances, might enhance model accuracy for extreme stellar types. Addressing these factors could provide a more holistic approach to astrophysical parameter estimation in future work.

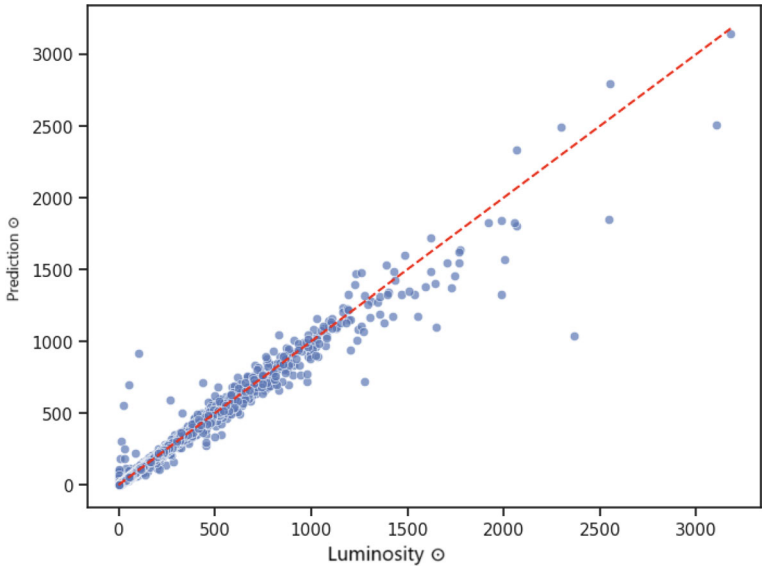


Fig. 11 Scatter plot of luminosity predictions using Random Forest

3.6 Methodological Insights and Future Directions

The methodological framework applied in this study demonstrates the effectiveness of machine learning models in estimating astrophysical parameters from extensive photometric data. Key aspects of data selection, feature engineering, and model choice have contributed significantly to the model's performance:

Data Selection and Cleaning: Ensuring high data quality was critical. The magnitude cutoff at 23 minimized observational noise, while incorporating Gaia's astrometric data expanded the feature set, enriched the dataset for more robust parameter estimation.

Feature Engineering and Model Selection: Constructed color indices and selected photometric features, particularly for effective temperature, radius, and luminosity estimations, significantly boosted model accuracy. Ensemble methods like XGBoost and Random Forest were chosen due to their strength in capturing complex relationships within high-dimensional data. In particular, L_1 and L_2 regularization techniques in XGBoost helped prevent overfitting by penalizing model complexity, thereby improving the generalizability of temperature estimations [15].

Potential Role of Quantum Computing: The application of quantum computing in machine learning has the potential to revolutionize astrophysical data analysis by significantly boosting computational efficiency. Quantum algorithms, such as quantum support vector machines or variational quantum algorithms, are particularly promising for handling high-dimensional and complex datasets like J-PLUS, as they can theoretically achieve polynomial or exponential speed-ups compared to classical methods [19]. In the context of large-scale astronomical datasets, quantum machine learning could enable faster parameter estimations and more accurate models, especially when combined with techniques like transfer learning to leverage pre-trained models on quantum hardware.

3.7 Implications and Opportunities for Future Research

The automation of stellar parameter extraction and estimation highlights machine learning's potential to streamline large-scale astrophysical analysis. Looking forward, several research opportunities emerge:

Integrating Quantum Machine Learning: As quantum computing technology matures, it can become a vital tool for astrophysics. Quantum-enhanced models show promise for processing and analyzing the vast datasets generated by surveys like J-PLUS more efficiently than traditional algorithms, enabling rapid and precise parameter estimation at scales currently unachievable [20]. Early-stage implementations of quantum algorithms may focus on tasks like dimensionality reduction or clustering, complementing and accelerating current methods in astrophysics.

Expanding Feature Sets: Parameter accuracy could be further improved by integrating additional data, such as spectral and temporal information, or by exploring

more complex architectures like neural networks to capture additional stellar characteristics. These architectures may offer more nuanced insights into stellar evolution and the chemical composition of stars.

4 Conclusions

This chapter explored the integration of artificial intelligence (AI) techniques in astrophysics, specifically focusing on the estimation of stellar parameters through machine learning models. The findings illustrate AI's potential to handle and analyze vast astronomical datasets efficiently, paving the way for automated, large-scale studies in astrophysics.

4.1 Summary of Findings

Key findings from this research include:

Effective Temperature Estimation: The XGBoost model achieved an R^2= 0.95 for effective temperature estimation, showcasing its ability to leverage photometric data to predict temperature with high precision. The model's use of color indices and other photometric features aligns well with astrophysical principles that relate color to temperature, reinforcing the model's interpretability.

Stellar Radius Estimation: XGBoost was also employed to estimate stellar radius, yielding an $R^2 = 0.98$. Luminosity and temperature emerged as the most influential features in this estimation, consistent with theoretical expectations based on the Stefan-Boltzmann Law. This result confirms the model's efficacy in incorporating key astrophysical variables to accurately predict radius.

Luminosity Estimation: The Random Forest model attained an $R^2 = 0.92$ for luminosity estimation, effectively capturing the intricate relationships between photometric data and stellar luminosity. The feature importance analysis highlighted parallax as the most impactful feature, aligning with its theoretical significance in estimating distance-dependent parameters like luminosity.

4.2 Contributions to the Field

This study contributes to the growing field of AI in astrophysics by providing a reproducible framework for estimating critical stellar parameters using photometric data. The integration of data from multiple catalogs, such as J-PLUS and Gaia, underscores the value of data fusion in enhancing model accuracy and expanding the scope of astrophysical analysis.

These findings emphasize the importance of machine learning as a tool for astrophysics, capable of reducing the manual workload involved in data analysis and enabling more efficient exploration of large astronomical datasets. Future research could build on this foundation by incorporating more comprehensive data sources, improving model architectures, and adapting this framework for broader astrophysical applications.

Acknowledgements This chapter has been made possible thanks to the support of project SIP20241558. We are deeply grateful to the Instituto Politécnico Nacional for their support and to the collaborators who contributed their expertise and dedication throughout the various stages of this research.

References

1. Angelis, A., Tatischeff, V., Tavani, M., *et al.*: The e-ASTROGAM mission. Journal of High Energy Astrophysics **19**, 1–106 (2018)
2. Cenarro, A.J., *et al.*: J-PLUS: A javalambre photometric local universe survey. Astronomy & Astrophysics **622**, 176 (2019)
3. Gaia Collaboration, Brown, A.G.A., Vallenari, A., Prusti, T., *et al.*: Gaia Data Release 1. Summary of the astrometric, photometric, and survey properties. Astronomy & Astrophysics **595**, 2 (2016)
4. Bonoli, S., et al.: The javalambre-physics of the accelerating universe astrophysical survey. Astronomy & Astrophysics **627** (2019)
5. Bailer-Jones, C.A.L.: Bayesian inference on stellar distances and extinctions using parallaxes, photometry, and models. Astronomy & Astrophysics **616**, 37 (2018)
6. Carroll, B.W., Ostlie, D.A.: An Introduction to Modern Astrophysics. Addison-Wesley, Reading, MA (1996)
7. Sánchez, L.H., Ojeda, A.V., Tapia, J., Guerrero, C.A.: Estimation of stellar parameters in j-plus dr3 through machine learning techniques. In: Iftekharuddin, K.M., Awwal, A.A.S., Diaz-Ramirez, V.H., Márquez, A. (eds.) Optics and Photonics for Information Processing XVIII, p. 131360. SPIE, Bellingham, WA (2024). https://doi.org/10.1117/12.3028734
8. Ivezic, Z., Connolly, A.J., VanderPlas, J.T., Gray, A.: Statistics, Data Mining, and Machine Learning in Astronomy. Princeton University Press, Princeton, NJ (2014)
9. Binney, J., Merrifield, M.: Galactic Astronomy. Princeton University Press, Princeton, NJ (2011)
10. López-Sanjuan, C., et al.: J-PLUS: Photometric calibration of large-area multi-filter surveys with stellar and white dwarf loci. EDP Sciences **A119** (2019)
11. Schlegel, D.J., Finkbeiner, D.P., Davis, M.: Maps of dust infrared emission for use in estimation of reddening and cosmic microwave background radiation foregrounds. The Astrophysical Journal **500**(2), 525–553 (1998)
12. York, D.G., *et al.*: The sloan digital sky survey: Technical summary. The Astronomical Journal **120**(3), 1579–1587 (2000)
13. Gray, R.O., Corbally, C.J.: Stellar Spectral Classification. Princeton University Press, Princeton, NJ (2009)
14. Gaia Collaboration, Brown, A.G.A., Vallenari, A., Prusti, T., de Bruijne, J.H.J., Babusiaux, C., Biermann, M.: Gaia Early Data Release 3: Summary of the contents and survey properties. Astronomy & Astrophysics **649**, 1 (2021) https://doi.org/10.1051/0004-6361/202039657
15. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, New York, NY (2016)

16. Bertin, E., Arnouts, S.: SExtractor: Software for source extraction. Astronomy and Astrophysics Supplement Series **117**(2), 393–404 (1996) https://doi.org/10.1051/aas:1996164
17. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
18. Karttunen, H., Kröger, P., Oja, H., Poutanen, M., Donner, K.J.: Fundamental Astronomy. Springer, Berlin, Germany (2017)
19. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
20. Preskill, J.: Quantum computing in the nisq era and beyond. Quantum **2**, 79 (2018)

Towards a Fully Quantum Learning System: Quantum Architecture Search with Quantum Evolutionary Algorithms

Yoshio Rubio, Cynthia Olvera, and Oscar Montiel

Abstract Similar to traditional neural architectures, quantum variational circuits face a key limitation in their bespoke, problem-specific designs. Inspired by neural architecture search, quantum architecture search aims to optimize gates and structures within these circuits. This study introduces an innovative quantum search algorithm tailored for NISQ devices which focuses on minimizing the search space, quantum circuit depth and size, while preserving accuracy. The proposal consists of a quantum evolutionary algorithm aided with local search that selects the best performing individuals in a supernet pool. Results exhibit a reduction in the computational footprint, while preserving accuracy.

Keywords Quantum computing · Quantum architecture search · Evolutionary algorithms · Quantum local search

1 Introduction

While the promise of quantum advantage has been a driving force behind decades of research in quantum computing [1, 2], it also presents significant challenges, including qubit stability, error correction, and hardware development [3]. In 2019, Google announced what was considered the first experimental evidence of quantum advantage [4], though more recent studies suggest that classical systems could achieve comparable performance with lower energy consumption and resources [5].

Y. Rubio, C. Olvera and O. Montiel—Equal contribution.

Y. Rubio (✉) · C. Olvera · O. Montiel
Centro de Investigación y Desarrollo de Tecnología Digital, Instituto Politécnico Nacional, Tijuana, Baja California, Mexico
e-mail: rrubio@citedi.mx

C. Olvera
e-mail: colvera@citedi.mx

O. Montiel
e-mail: oross@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_20

Significant advancements in recent years have resulted in the development of Noisy Intermediate-Scale Quantum (NISQ) devices, which are quantum processors with a limited number of qubits and no fault tolerance [6]. Although these devices can outperform classical computers in specific, artificially designed tasks, they currently lack practical application. The pressing challenge is to develop algorithms that leverage the capabilities of NISQ devices to achieve quantum advantage in tasks of real-world relevance [7].

Variational quantum algorithms (VQAs) have emerged as a promising class of algorithms for NISQ devices due to their robustness against errors and their ability to operate on hardware with limited qubit connectivity and shallow circuit depth [8]. VQAs are often considered the quantum analog of neural networks, making them well-suited for a variety of machine learning tasks [2]. VQAs have been applied in various applications and domains, such as quantum classification [9–11], generative modeling [12, 13], quantum information analysis [14, 15], quantum data compression [16, 17], and ground states [18] and excited states [19] search.

VQAs are powered by parameterized quantum circuits (PQCs) [8, 20], which utilize learnable parameters consisting of both fixed and adjustable gates (typically rotation gates). These parameters are optimized by a classical optimizer to minimize an objective function, enabling VQAs to tackle complex quantum problems [2, 8, 21]. However, one of the most significant challenges in PQCs is the occurrence of barren plateaus, where the gradients vanish exponentially, making the optimization process inefficient. Recent research has shown that this issue can be mitigated through the designed of carefully crafted, quasi-optimal circuits [22]. Nevertheless, manually designing such circuits is a highly complex and resource-intensive task, making it a persistent challenge and a focal point of ongoing research [20].

To face these challenges, quantum architecture search (QAS) has emerged as an automated approach to efficiently generate optimized circuit designs. QAS aims to automatically find the best PQC from a vast set of possible architectures, providing an advantage over traditional trial-and-error methods [20]. Strategies for identifying the optimal ansatz vary widely, from brute-force approaches to more advanced intelligent search techniques, such as evolutionary algorithms and reinforcement learning [20, 23]. The computational costs associated with these methods can be significant, as they scale with both the depth and number of qubits in the circuit. Additionally, the overall cost is influenced by the number of search iterations and optimization epochs required to converge on a solution.

Our proposal focuses on balancing the exploration and exploitation of the search space using global and local search. The method uses a quantum evolutionary algorithm that focuses on quantum local search that performs well under limited populations and epochs. The search is done in a pool of possible architectures encoded in a supernet. Our proposal aims to reduce quantum footprint while performing similarly to other QAS that perform extensive search steps. The contribution of this work is as follows:

- We propose a novel *evolutionary search strategy* that integrates within a hybrid quantum-classical architecture. This component, named Quantum Local Evolu-

tionary Algorithm (QLEA), operates on the classical side, where it guides the architectural exploration by selecting and refining candidate quantum anzatz based on their performance, as evaluated through quantum circuit executions.
- Building upon this strategy, we introduce the Local Evolution Quantum Architecture Search (LEQAS), a hybrid framework that incorporates QLEA to accelerate convergence. LEQAS requires fewer evolutionary steps than traditional quantum architecture search methods, while maintaining competitive performance.

This work is structured as the following: the definition of the theoretical background and the proposed method is described in Sect. 2; followed by the experiments and discussion of the results in Sect. 3; and the conclusions in Sect. 4.

2 Methods

In the following subsections we describe the fundamentals of variational quantum algorithms, quantum architecture search, as well as the proposed method in detail.

2.1 Variational Quantum Algorithm

Although VQAs have some variations in their algorithmic structure, there are shared elements between different VQAs. Figure 1 defines the common steps to design VQAs. The first step in designing a VQA involves defining a loss or cost function C,

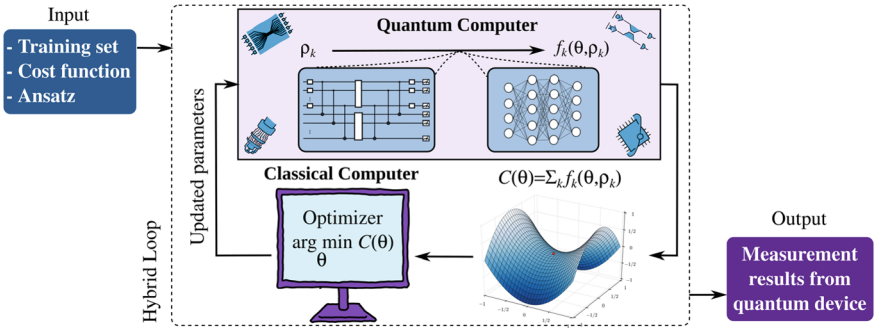


Fig. 1 VQA diagram. In a VQA, the inputs are the training set, a cost function $C(\theta)$ with θ parameters that encode the solution to the problem, and an ansatz whose parameters are optimized to minize the cost function. The ansatz can be described as a parametric circuit or as an analogous to a deep neural network. In each iteration of the hybrid loop, the quantum device estimates the cost function and the information is fed into a classical system to optimize the cost function. The final output of the system can be a quantum state, a probability distribution, a bit string, or any other useful output for a given problem

which is used to evaluate the performance of the quantum circuit in addressing the given problem. This function quantifies the discrepancy between the circuit's output and the desired outcome, thereby guiding the optimization of the circuit's parameters towards an effective solution.

The next step is to propose a PQC (also called ansatz), which is a quantum operation that depends on a set of continuous or discrete optimizable parameters θ. The parameters in the ansatz are trained to solve the optimization task [2]:

$$\theta^* = \arg\min_{\theta} C(\theta). \tag{1}$$

The PQC optimization is a hybrid quantum-classical process, since the estimation of the cost function $C(\theta)$, or gradient calculation, is done in quantum device, while the optimization of the parameters θ is done in a classical device.

As previously mentioned, the main core of a VQA is the ansatz. The design of an ansatz can be problem-specific, or it can be general-purpose. The unitary gates are commonly encoded into block patters, commonly called layers, which can be manually or automatically configured. In automatic methods (such as QAS) all the available unitary gates and combinations between these gates are put into a pool and the automatic method will configure the best layer based on a set of restrictions.

To calculate the cost function, the fixed and trainable parameters in an ansatz are encoded into unitary $U(\theta)$, which is applied to the input states of the circuit, see Fig. 2. The unitary $U(\theta)$ can be expressed as the product of L sequentially applied unitaries (layers):

$$U(\theta) = U_L(\theta_L) \cdots U_2(\theta_2) U_1(\theta_1), \tag{2}$$

in which

$$U_l(\theta_l) = \prod_m e^{-i\theta_m H_m} W_m, \tag{3}$$

where $e^{-i\theta_m H_m}$ represents the parametric component with Hermitian operator H_m, and W_m is the unparametrized component [2].

Once the components of $U(\theta)$ are set, and the cost function is defined, the next step is to find the optimal θ parameters. The most common method for gradient calculation is using the parameter-shift rule. For this method let us consider a cost function with $f_k(x) = x$, and θ_l be the l-th parametric element in the unitary $e^{i\theta_l \sigma_l}$, and σ_l be a Pauli operator. The parameter-shift rules state that the gradient of the cost is equal to:

$$\frac{\partial C(\theta)}{\partial \theta_l} = \sum_k \frac{1}{2 \sin \alpha} \mathrm{Tr}[B_k U^\dagger(\theta_+) \rho_k U(\theta_+)] - \mathrm{Tr}[B_k U^\dagger(\theta_-) \rho_k U(\theta_-)] \tag{4}$$

where ρ_k is the k-th mixed state $\rho_k = \sum_i p_i |\psi_i\rangle\langle\psi|$; p_i are the eigenvalues of ρ such that $\sum_i p_i = 1$; $\theta_\pm = \theta \pm \alpha e_l$ holds for any real number α; e_l is a vector of 1 in the l-th parameter; and B is an observable [2, 24–26].

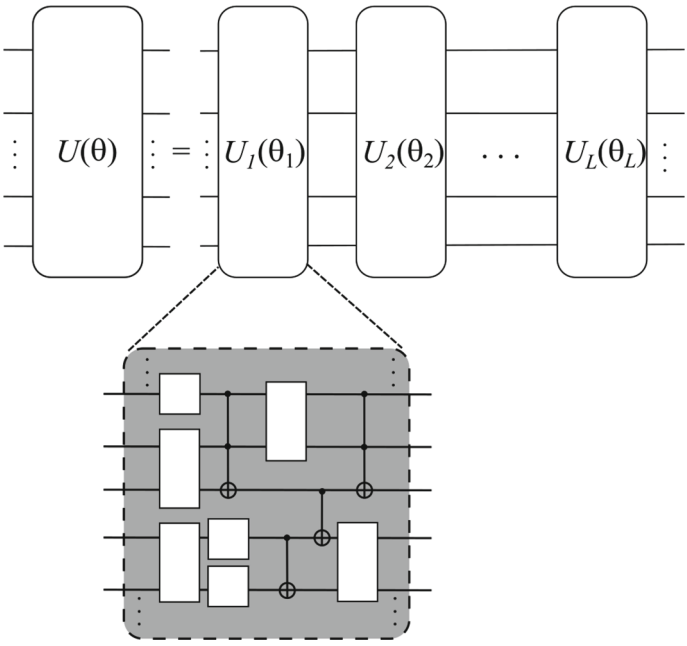


Fig. 2 Ansatz diagram. The ansatz is encoded into $U(\theta)$ with trainable and fixed parameters θ. The unitary $U(\theta)$ can be expressed as the product of L unitaries $U(\theta_l)$, which can be decomposed into a given set of parametrized and unparametrized gates. In the block we can see a combination of gates that require one, two or three quantum bits

With the parameter-shift rule the gradient of the cost with respect each l-th trainable parameter can be calculated and used to update the circuit with any selected optimizer.

2.2 Quantum Architecture Search

The aim of QAS is to automatically design a PQC for a specific task to be solved using an specified performance criteria and a given set of restrictions [20]. Quantum architecture search has a strong resemblance to neural architecture search (NAS), which aims to automatically generate neural networks architecture for a given set or restrictions and objectives.

In QAS, the search space describes all the possible architectures that can be constructed based on the architectural restrictions and available unitaries for $U(\theta)$. Commonly, the available gates and granularity are defined based on the design requirements, the given tasks, and other limitations (such as hardware restrictions). The

granularity may include gate, layer, and block level configuration, as well as restrictions based on the number of qubits.

To represent a circuit into a search space, QAS rely on different encoding techniques. One of the most common approaches is the list representation, which encodes quantum gates along with relevant data such as positional information [20].

Additionally, other two important aspects for QAS are the search strategy and the selection of a performance evaluation method. The search strategy describes the approach to explore the search space to find the optimal architecture, while the performance evaluation describes the process of measuring the fitness of the proposed circuits during the search strategy [20].

2.3 Proposed Method

The proposed method is an expansion over [23] in which the authors use a supernet to encode the pool of architectures, optimize a sampled set of anzats, rank the set, and perform additional training of the set. Our proposal aims to reduce the amount of optimization of the sampled set by creating a quantum population that evolves locally to the direction of the fittest members of the population at each generation.

With our proposal, called Local Evolution Quantum Architecture Search (LEQAS), we aim for local and global exploration at different configurable granularity. The main engine of LQAS is the quantum local evolutionary algorithm (QLEA), which evolves the population into the direction of the best members of the population, with a configurable amount of change that focuses on local searches. LEQAS has eight main steps:

1. Generate a *supernet* that contains the pool of all available architectures with weight sharing approach.
2. Select a set of random architectures and encode them using a quantum population with n individuals and m experts (random set of shared weights).
3. Select the best expert per individual and rank population.
4. Replace the expert of each individual with the expert of the best individual.
5. Evaluate the population with the best expert and rank population.
6. Perform quantum local search using QLEA, with the best individuals as parents, and the population every generation.
7. Optimize each individual using *parameter-shift* rule with a limited set of epochs.
8. Select the best individual, and perform optimization over a fixed number of epochs with parameter-shift rule.

In the next subsection we describe in detail each major component and step of LEQAS.

2.3.1 Supernet

Similarly to [23], the proposed method relies on a supernet that encompases all the possible architectures in the search space and its submodels (ansatz pool). The advantage of supernets is that weight sharing reduce computational complexity while retaining good performance [27].

The setup of the supernet is focused on tracking the ansatz pool with linear memory cost. To achieve this, each possible permutation of quantum gate of each layer U_L, is stored based on the number of available gate and number of qubits, see Fig. 3. For our approach we limited the layer to include single qubit gates $\{R_X, R_Y, R_Z\}$ and two-qubit gates $\{CX, CY, CZ\}$ (controlled quantum gates). For our implementation, the number of qubits of the circuit is given by the number of input features given by the classification task.

The tracking of the possible ansatz is done using a list of integer values for each possible permutation, in which component will be a layer U_L. As an example, lets assume that we have 2 possible rotation gates $\{R_X, R_Y\}$, and three layers. The possible permutations (with repetitions) should be $\{(R_X, R_X), (R_X, R_Y), (R_Y, R_X), (R_Y, R_Y)\}$. An ansatz with architecture [1, 3, 0] can be expressed as $U(\theta) = (R_X \otimes R_Y)(R_Y \otimes R_Y)(R_X \otimes R_X)$.

Additionally, the supernet has an encoding layer which are used to represent the input values into quantum states in the circuit. For our purposes, the normalized values are encoded with a Hadamard gate, followed by angle encoding with R_y, and finalized by a controlled rotation gate in the y-axis between adjacent qubits between adyacent wires, see Fig. 4. The rotation angle of the controlled gate can be defined based on the use case.

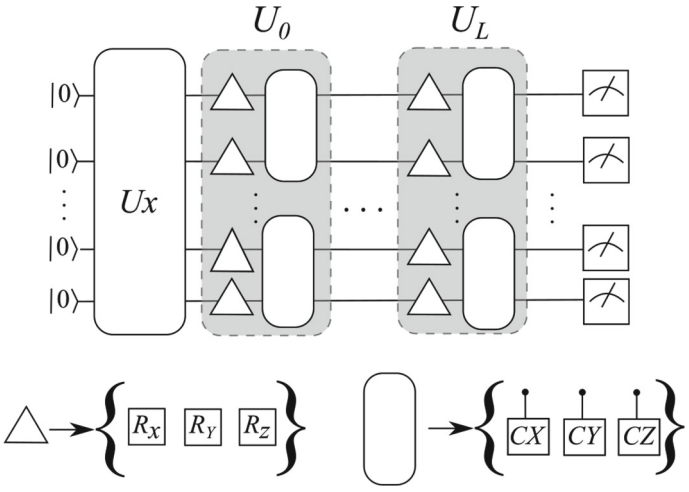


Fig. 3 The supernet consists of a quantum encoding block followed by L layers. For each layer, the available single-qubit gate are quantum rotations R_X, R_Y, R_z; and two-qubit gates are CX, CY, CZ

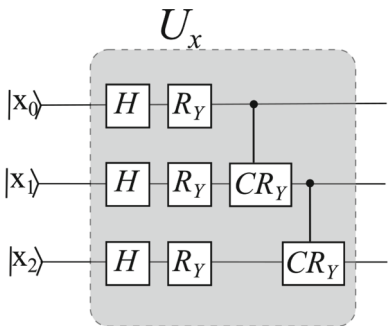


Fig. 4 Encoding layer for a three-qubit input. The encoding layer consists of a Hadamard for each input, followed by a R_Y gate that encodes the input feature with angle encoding. Following, are two controlled rotations in the y-axis that act between adjacent wires

2.3.2 Quantum Local Evolutionary Algorithm

As previously mentioned, the complete search space for the supernet is defined by the depth or number of layers of the supernet (L), the number of q qubits (given by the number of input features), and the number of available quantum gates (designed by the human expert).

In our proposal, after defining the supernet, we create a quantum population $Q(t)$ to encode a set of N quantum individuals. The individuals of the quantum population are encoded into a single qubit per layer, making each quantum chromosome encodes an specific architecture in the pool. The i-th individual at any given time can be defined as:

$$|\psi^i(t)\rangle = |\psi_0^i(t)\rangle|\psi_1^i(t)\rangle \cdots |\psi_L^i(t)\rangle, \tag{5}$$

using Eq. 5 as a reference we can define the quantum population as:

$$Q(t) = [|\psi^0(t)\rangle, |\psi^1(t)\rangle, \ldots, |\psi^N(t)\rangle] \tag{6}$$

Before initializing the quantum population, the shared weights for the supernet are randomly generated based on the number of parametric values. As suggested by [23], G experts with random value are created per individual with the aim of improving diversity. Each expert g will contain a separated and isolated set of initial weights for the supernet.

In the quantum individual, the permutation value of the selected layer is encoded in the qubit using $R_Y(\omega)$, in which $\omega \in [0, \pi/2]$, and ω is calculated by normalizing the permutation integer of the layer in the aforementioned range. Since the index of the layer is an integer value, when converting from angle to index, we round up to the closest integer.

At $t = 0$, the initial quantum population $Q(t)$ is randomly generated. Each individual $\psi^i(t)$ is decoded to create the current ansatz a^i. The parametric weights of a^i are initialized by each expert of the individual. The encoding layer of the ansatz is initialized using a set of inputs features $\boldsymbol{x}$, and the circuit is measured by calculating the expectation value $\langle A \rangle$ of the last layer U_l. The expectation value $\langle A \rangle$ is used to

generate an estimation of the output $\hat{y}$. The fitness of expert g-th expert for the i-th individual is calculated using a loss function $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

The fitness of each quantum individual is calculated using each expert and the best expert is selected as candidate shared weights. The population is then ranked based on its fitness, and the best expert of the best individual is selected to be used as the shared weights. The population is then evaluated again and ranked.

The next step is to perform a quantum local search for some members in the population trying to move closer to the best members of the population. At each search step, the quantum population will be sorted based on fitness, and the best b elements will be put into a pool of parents, and the rest will be put into an evolution pool. Each k-th individual in the evolution pool will be compared at a qubit level with a random parent p_k and updated in the direction of the encoding angle:

$$\begin{cases} w_k^i = w_k^i - \Delta_\theta, & \text{if } w_p^i < w_k^i \\ w_k^i = w_k^i + \Delta_\theta, & \text{otherwise} \end{cases} \tag{7}$$

in which Δ_θ will be the step size to perform the local search. After updating each quantum individual, the population will be updated and ranked. With this approach, local search can provide quantum individuals that have a higher fitness at each search step.

2.3.3 Quantum Population Optimization

After the quantum local search, each quantum individual is optimized using *parameter-shift* rule for a given set of steps. Parameter-shift is commonly used as the standard method to optimize variational autoencoders. To avoid issues due to weight-sharing, the initial weights will be the same for each quantum individual instead of starting from the optimized set of weights of the previous individual in the population.

Once all the individuals have been optimized, the best individual is selected as the final ansatz and an extensive optimization will be done with this individual.

3 Results and Discussion

To test the capacity of the proposed method we first performed an ablation study using the synthetic dataset defined in [23]. The dataset is for binary classification and has 300 samples (100 for training, 100 for test, and 100 for validation). The base case using LQEAS has a quantum population of 10 individuals, 3 qubits per input features, 4 quantum layers, 5 experts per individual, 5 QLEA steps, and was trained during 10 epochs with a learning rate of 0.05 and Adam optimizer. The rotation angle for the controlled gate in the encoder was set as $\Pi^l(\pi - x_l)$ for each feature. For the

Table 1 Accuracy of the base case and the six cases in the ablation study. The model with highest test accuracy was experiment III, which 5 quantum layers

Accuracy	Base	I	II	III	IV	V	VI
Train	0.84	0.8	0.75	0.86	0.85	0.8	0.85
Test	0.86	0.75	0.76	0.86	0.83	0.77	0.82

ablation experiments, we used the base case as reference but with modifications to specific parameters:

I. Epochs: 5
II. Experts: 10
III. Layers: 5;
IV. Population: 20; Layers: 5
V. Epochs: 5; Population: 20
VI. Population: 20.

The results of the best architecture found using the base case and the results of the best architecture of each case in the ablation study are described on Table 1. From the ablation, we can observe that the best performance was achieved by the third experiment (5 quantum layers), it has the same performance in the test set as the base case, but it has higher performance in the training set. The worst experiment was Case II, in which the increase of experts affected the performance of the whole population. The cases with 20 individuals behaved similarly between them except for Case V in which the number of epochs was decreased. The other experiment with 5 epochs (Case 1) also had low performance in the test set, which reflects that using 10 epochs is the minimum number of epochs to have accuracy higher tha 0.8 in the test set.

The evolution of the base case and the six ablation experiments can be seen from Figs. 5 and 6f, in these Figures the title indicates (in order): the number of quantum individuals, qubits, quantum layers, experts, and training epochs. In the best configuration, Fig. 5, the best individual for the training set at t_0 had an accuracy below the other top-3 individuals (0.51), and decreased when the best expert was implemented in t_1. It can be seen that the increase in the accuracy (0.82) happens in t_2 after 1 QLEA epoch and when optimization step began. Individual-2 also had 1 QLEA epoch but with more discrete improvement. It must be noted that the best test performance was achieved by individual-2, that had a slowly and increase in the accuracy during the optimization stage.

Analyzing the other Cases: I, II and VI, QLEA enabled some quantum individuals to improve its performance. In Case II, individual-2 had 5 epochs of QLEA and had the highest training accuracy, followed by individual-1 who had 1 QLEA epoch. In a similar sense, in Case VI, individual-1 had 5 QLEA epochs and achieved the second highest training accuracy for that experiment.

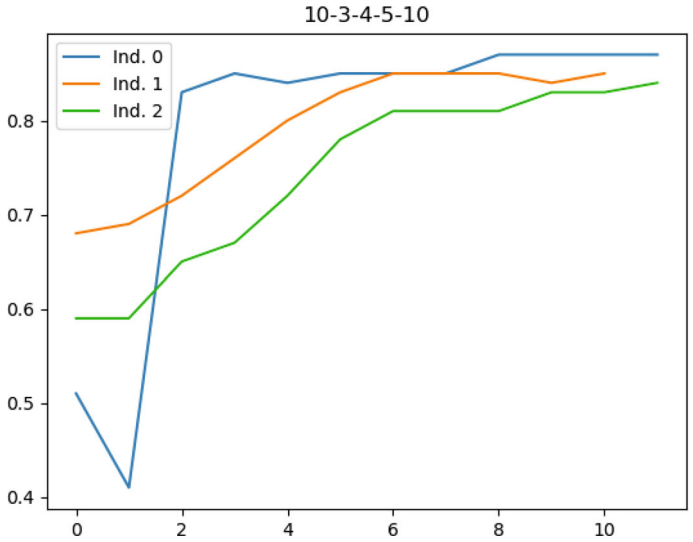
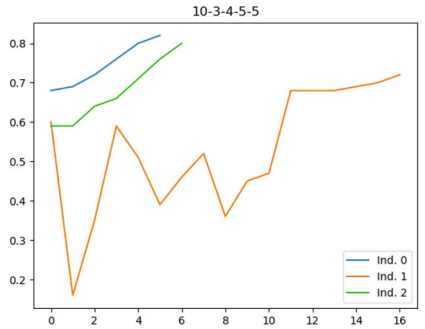


Fig. 5 Base case: 10 individuals, 3 qubits, 4 layers, 5 experts and 10 epochs, x-axis epochs, y-axis accuracy. In this evolution the best individual began very far from the other individuals, but QLEA enabled the quantum individual to surpass the two best members at t_2

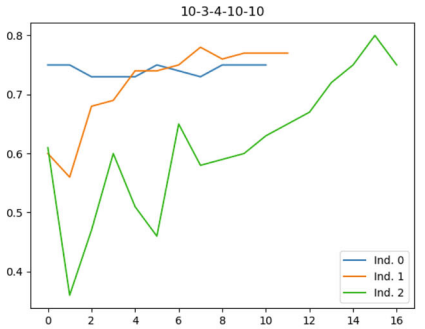
We compared the performance of LEQAS against the QAS proposed in [23]. For this experiment we used the same proposed parameter as the authors in QAS: 400 epochs, 500 searches, 5 experts, 3 qubits, and 3 quantum layers. From the search, the five best architectures were selected and trained from scratch 400 epochs. For the comparisons use the best circuit found by LEQAS and derived three models: the model trained during the search (A), the base model trained after the search with the same set of weights with additional 100 epochs (B), and the base model trained after the search for 100 epochs with random set of weights (C). The results can be seen in Table 2.

The results show that all the QAS architectures had a training accuracy during search <=0.1 compared to base LEQAS. If we trained for additional epochs, the best QAS architecture [64,190,180] reaches the same training accuracy that LEQAS, but three of the remaining four had a decrease in training performance since they start from a random set of weights. The biggest difference comes with the test accuracy, the base LEQAS had the highest performance before further optimization steps (cases B and C). Compared to the best QAS architecture in the test set LEQAS had a difference of 0.07 in the test set. The best architecture obtained in with the LEQAS method can be seen in Fig. 7, while the best architecture found with QAS can be seen in Fig. 8.

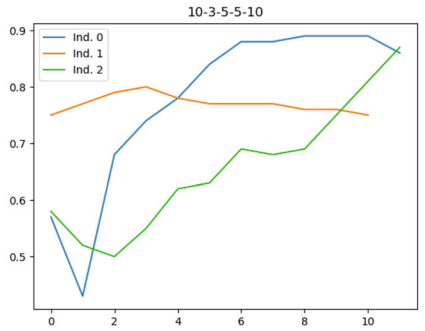
The main advantage of the proposed method is that QLEA is able to mix local and global search due to its inherent properties. Results shown in Tables 1, 2 show that LEQAS achieve better performance in the test set using less searches in the search space than QAS. Using a reduced population, while QAS needs 500 searches plus



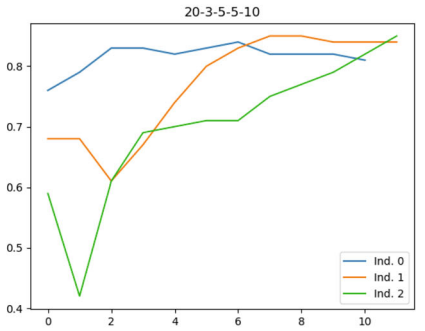
(a) Case I



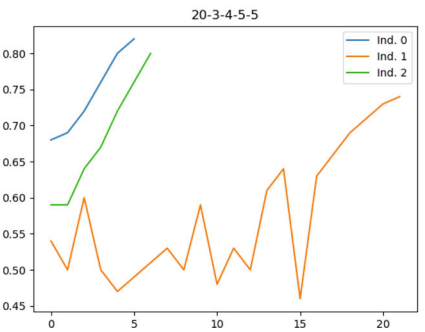
(b) Case II



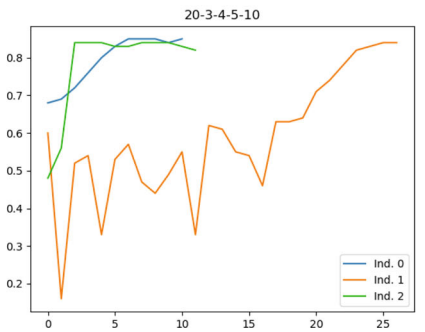
(c) Case III



(d) Case IV



(e) Case V



(f) Case VI

Fig. 6 Six ablation experiments, for all the figures the x-axis is the epochs, and the y-axis accuracy. From case, I, II, III, IV, and VI we can observe the relevance of the method. Although, the quantum evolutionary algorithm is able to find a good candidate solution using quantum local search, the last steps to improve the performance of the method with gradient optimization helps on improving solutions that were not the best in the initial generations. Additionally, the proposed method is able to find suitable solutions less than 25 generations

Table 2 Comparisons of the best five architectures found by QAS (first five rows) and three variants of the best architecture found with LEQAS. Train acc (I) indicates the training accuracy during search, and (II) indicates the training accuracy after additional optimization epochs

Architecture	Train acc (I)	Train acc (II)	Test acc
[6,77,56]	0.83	0.75	0.68
[119,158,152]	0.83	0.75	0.71
[209,170,180]	0.81	0.81	0.78
[64,190,180]	0.83	0.87	0.79
[30,169,8]	0.77	0.78	0.79
LEQAS (A)	0.84	N/A	0.86
LEQAS (B)	0.84	0.87	0.8
LEQAS (C)	0.84	0.86	0.8

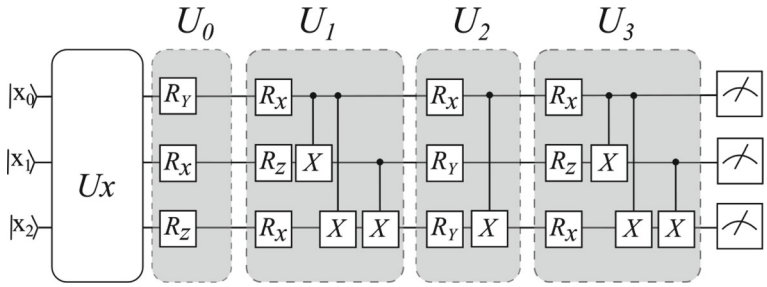


Fig. 7 Best ansatz found with LEQAS

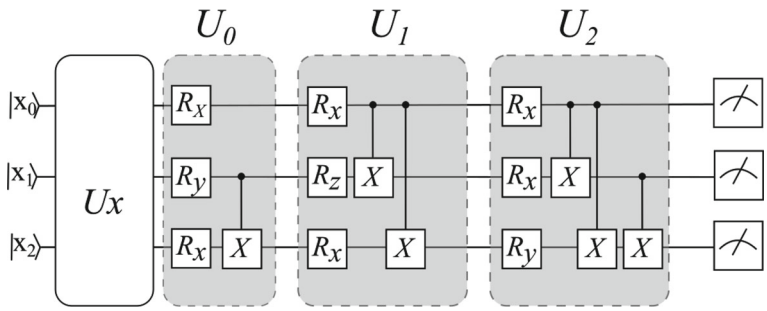


Fig. 8 Best ansatz found with QAS

800 optimization steps to achieve similar results in train set, LEQAS achieves this in less than 15 steps (combination of five QLEA steps and 10 optimization epochs). Additionally, its performance was better in the test set with just 10 optimization steps. This is crucial in quantum devices in which the computation resources are limited.

4 Conclusion

In this work we proposed a quantum architecture search method, that achieves similar performance as state-of-the-art methods with a fraction of the search steps. The method has as a core a quantum local search algorithm named QLEA which encodes a quantum population of a set of qubits per quantum individual and evolves it to improve the performance of the population. In QLEA, a member of the fittest individual is selected and used as a reference to change the values of the quantum bits of a less fit member of the population.

Our experiments in the selected training and test set show that the proposed LEQAS achieves 10% higher accuracy in the test set compared to the QAS method. Additionally, the compared QAS needed 40 times more searches in the supernet pool to achieve similar performance in the training and test sets.

The proposed method not only achieves higher performance in less computation steps than the compared QAS, but it also limits the amount of optimization steps in the classical part of the VQA hybrid classical-quantum system, relying more in the quantum local search steps. This is a move into a direction in which the quantum steps in the hybrid quantum-classical system can have higher relevance than the classical steps.

Although there are limitations in the current results since they only show the performance in a synthetic and limited dataset, it is a step in the right direction on developing algorithms for quantum architecture search and VQAs that can develop quantum circuits for more practical use cases. Future work can focus on more complex problems such as image classification or more than two categories classification.

Acknowledgements The research leading to these results has received funding from the Instituto Politécnico Nacional (IPN), under Research Projects SIP20240164, and the Consejo Nacional de Humanidades Ciencias y Tecnologias (CONAHCYT) through project CF-2023-I-108.

References

1. Clinton, L., Bausch, J., Cubitt, T.: Hamiltonian simulation algorithms for near-term quantum hardware. Nature Communications **12**, 4989 (2021) https://doi.org/10.1038/s41467-021-25196-0
2. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., Coles1, P.J.: Variational quantum algorithms. Nature Reviews Physics **3**, 625–644 (2021) https://doi.org/10.1038/s42254-021-00348-9
3. Meirom, D., Alfassi, I., Mor, T.: Discretized Quantum Exhaustive Search for Variational Quantum Algorithms (2024). https://arxiv.org/abs/2407.17659
4. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M.P., Hartmann, M.J., Ho, A., Hoffmann, M., Huang, T., Humble, T.S., Isakov, E. Sergei V.and Jeffrey, Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P.V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark,

M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M.Y., Ostby, E., Petukhov, A., Platt, J.C., Quintana, C., Rieffel, E.G., Roushan, P., Rubin, N.C., Sank, K.J. Daniel Satzinger, Smelyanskiy, V., Sung, K.J., Trevithick, M.D., Vainsencher, A., Villalonga, B., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Neven, H., Martinis, J.M.: Quantum supremacy using a programmable superconducting processor. Nature **574** (2019) https://doi.org/10.1038/s41586-019-1666-5

5. Fu, R., Su, Z., Zhong, H.-S., Zhao, X., Zhang, J., Pan, F., Zhang, P., Zhao, X., Chen, M.-C., Lu, C.-Y., Pan, J.-W., Pei, Z., Zhang, X., Ouyang, W.: Achieving Energetic Superiority Through System-Level Quantum Circuit Simulation (2024). https://arxiv.org/abs/2407.00769
6. Preskill, J.: Quantum Computing in the NISQ era and beyond. Quantum **2**, 79 (2018) https://doi.org/10.22331/q-2018-08-06-79
7. Nicoli, K.A., Anders, C.J., Funcke, L., Hartung, T., Jansen, K., Kühn, S., Müller, K.-R., Stornati, P., Kessel, P., Nakajima, S.: Physics-Informed Bayesian Optimization of Variational Quantum Circuits (2024). https://arxiv.org/abs/2406.06150
8. He, Z., Deng, M., Zheng, S., Li, L., Situ, H.: Gsqas: Graph self-supervised quantum architecture search. Physica A: Statistical Mechanics and its Applications **630**, 129286 (2023) https://doi.org/10.1016/j.physa.2023.129286
9. Acampora, G., Chiatto, A., Vitiello, A.: Training circuit-based quantum classifiers through memetic algorithms. Pattern Recognition Letters **170**, 32–38 (2023) https://doi.org/10.1016/j.patrec.2023.04.008
10. Das, M., Bolisetti, T.: Variational Quantum Neural Networks (VQNNS) in Image Classification (2023)
11. Bai, Q., Hu, X.: Quantity study on a novel quantum neural network with alternately controlled gates for binary image classification. Quantum Information Processing **22**, 184 (2023) https://doi.org/10.1007/s11128-023-03929-y
12. Zhou, N.-R., Zhang, T.-F., Xie, X.-W., Wu, J.-Y.: Hybrid quantum-classical generative adversarial networks for image generation via learning discrete distribution. Signal Processing: Image Communication **110**, 116891 (2023) https://doi.org/10.1016/j.image.2022.116891
13. Ngo, T.A., Nguyen, T., Thang, T.C.: A survey of recent advances in quantum generative adversarial networks. Electronics **12**(4) (2023) https://doi.org/10.3390/electronics12040856
14. Zhao, X., Zhao, B., Wang, Z., Song, Z., Wang, X.: Practical distributed quantum information processing with loccnet. npj Quantum Information **7**(159) (2021) https://doi.org/10.1038/s41534-021-00496-x
15. Chen, R., Zhao, B., Wang, X.: Near-term efficient quantum algorithms for entanglement analysis. Phys. Rev. Appl. **20**, 024071 (2023) https://doi.org/10.1103/PhysRevApplied.20.024071
16. Pivoluska, M., Plesch, M.: Implementation of quantum compression on ibm quantum computers. Scientific Reports **12**(1), 5841 (2022) https://doi.org/10.1038/s41598-022-09881-8
17. Patel, S., Collis, B., Duong, W., Koch, D., Cutugno, M., Wessing, L., Alsing, P.: Information loss and run time from practical application of quantum data compression. Physica Scripta **98**(4), 045111 (2023) https://doi.org/10.1088/1402-4896/acc492
18. Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G.H., Tennyson, J.: The variational quantum eigensolver: A review of methods and best practices. Physics Reports **986**, 1–128 (2022) https://doi.org/10.1016/j.physrep.2022.08.003
19. Gocho, S., Nakamura, H., Kanno, S., Gao, Q., Kobayashi, T., Inagaki, T., Hatanaka, M.: Excited state calculations using variational quantum eigensolver with spin-restricted ansätze and automatically-adjusted constraints. npj Computational Materials **9**(13) (2023) https://doi.org/10.1038/s41524-023-00965-1
20. Martyniuk, D., Jung, J., Paschke, A.: Quantum Architecture Search: A Survey (2024). https://arxiv.org/abs/2406.06210
21. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quantum Science and Technology **4**(4), 043001 (2019) https://doi.org/10.1088/2058-9565/ab4eb5

22. Larocca, M., Thanasilp, S., Wang, S., Sharma, K., Biamonte, J., Coles, P.J., Cincio, L., McClean, J.R., Holmes, Z., Cerezo, M.: A Review of Barren Plateaus in Variational Quantum Computing (2024). https://arxiv.org/abs/2405.00781
23. Du, Y., Huang, T., You, S., Hsieh, M.-H., Tao, D.: Quantum circuit architecture search for variational quantum algorithms. npj Quantum Information **8** (2022) https://doi.org/10.1038/s41534-022-00570-y
24. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Physical Review A **98**(3) (2018) https://doi.org/10.1103/physreva.98.032309
25. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., Killoran, N.: Evaluating analytic gradients on quantum hardware. Physical Review A **99**(3) (2019) https://doi.org/10.1103/physreva.99.032331
26. Wierichs, D., Izaac, J., Wang, C., Lin, C.Y.-Y.: General parameter-shift rules for quantum gradients. Quantum **6**, 677 (2022) https://doi.org/10.22331/q-2022-03-30-677
27. Zhang, Y., Zhang, Q., Yang, Y.: How Does Supernet Help in Neural Architecture Search? (2021). https://arxiv.org/abs/2010.08219

Roach Infestation Optimization Algorithm with Enhanced Performance Using Automatic Parameter Adaptation Based on Fuzzy Systems

Enrique Lizárraga, Fevrier Valdez, Oscar Castillo, and Patricia Melin

Abstract Inspired by the social behavior of cockroaches, the Roach Infestation Optimization (RIO) algorithm was developed, utilizing three primary behaviors of cockroaches: seeking the darkest location, socializing with nearby cockroaches, and experiencing hunger, which prompts them to leave their comfort zones. These behaviors, combined with evolutionary algorithms, yield more satisfactory results. To further enhance RIO's performance, an adaptation of the C_{max} parameter representing the cognitive and social adjustment was introduced to balance exploration and exploitation, thereby improving algorithm convergence. As a result, the RIO algorithm with fuzzy adaptation outperformed the original RIO when evaluated using various benchmark mathematical functions.

Keywords Roach algorithm · Fuzzy logic · Optimization

1 Introduction

In a world of constant technological growth, the use of evolutionary algorithms to solve various types of problems has become increasingly popular. This trend has led to the development of numerous nature-inspired algorithms. One of the most well-known optimization algorithms is PSO (Particle Swarm Optimization), proposed by Eberhart and Kennedy in 1995 [10].

The RIO (Roach Infestation Optimization) algorithm, as a variant of PSO, applies the same fundamental principles. For this reason, it has been selected as the subject of study in this work. Introducing a fuzzy adapter that modifies certain parameters through fuzzy rules aims to improve the performance of the RIO algorithm. This adapter will adjust the parameters according to the most suitable rules, enhancing the algorithm's ability to effectively handle various types of problems [19].

E. Lizárraga · F. Valdez (✉) · O. Castillo · P. Melin
Tijuana Institute of Technology, TecNM, Tijuana, BC, México
e-mail: fevrier@tectijuana.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_21

Combining PSO with fuzzy logic through an adapter allows for dynamic and effective parameter adjustment. This technique offers the potential for more efficient solutions to a wide range of problems, significantly improving the performance of optimization algorithms like RIO [4].

Considering quantum computing in this context is crucial, as its implementation could revolutionize the way optimization problems in AI are addressed, leading to faster and more effective solutions. In this case, RIO algorithm enhanced with a fuzzy adapter, quantum computing would allow for the simultaneous evaluation of multiple parameter configurations, facilitating more efficient exploration and exploitation of the search space.

Several studies support the effectiveness of dynamic parameter adaptation using fuzzy logic in optimization algorithms. For example, [8] demonstrated interval type-2 fuzzy logic for parameter adaptation in particle swarm optimization. Similarly, [12] enhanced dynamic parameter adaptation in the Bird Swarm Algorithm, while [24] conducted a comparative study on fuzzy dynamic parameter adaptation in the Gorilla Troops Optimizer and Stochastic Fractal Search.

In this work, a fuzzy adapter was developed with 9 fuzzy rules to adjust the C_{max} parameter, thereby achieving a better balance between exploration and exploitation, avoiding combinatorial explosion within the algorithm and thus preventing the algorithm from overexploiting the solutions.

The structure of this article is as follows: In Sect. 2, the theoretical framework will be presented, discussing the context of evolutionary algorithms and their integration with fuzzy logic, where various improvements have been achieved through parameter adaptation. Subsequently, the development of the RIO algorithm and its most recent variants will be addressed. In Sect. 3, the proposed method is introduced. Section 4 will discuss the development of fuzzy adapters. In Sect. 5, the obtained results are presented to determine whether fuzzy parameter adaptation has improved the performance of the RIO algorithm.

2 Theoretical Framework

Optimization is the process of adjusting the inputs or features of a device, mathematical process, or experiment to find the minimum or maximum result. The inputs consist of variables, and the process or function is known as the cost function, objective function, or fitness function; the result is the cost or fitness. There are various methods for solving optimization problems, many of which are inspired by natural processes, and one of the most well-known is PSO (Particle Swarm Optimization) [2].

PSO has recently undergone many improvements and applications, with a primary focus on enhancing convergence and increasing the swarm's diversity. Hybrid and adaptive approaches have been proposed, such as a PSO based expert system integrated with fuzzy logic for the diagnosis of diseases, and an adaptive approach for function approximation. These advancements aim to address the limitations of

traditional PSO by dynamically adjusting key parameters to optimize performance across various scenarios, leading to more efficient and accurate results. The combination of PSO with fuzzy logic offers a promising direction for further improving its adaptability and robustness in solving complex problems [3].

Due to the remarkable effectiveness of the PSO algorithm in solving optimization problems, many variants have emerged, including the RIO algorithm. This method was proposed by Havens et al. [19].

It is inspired by the collective and individual behaviors of cockroaches, which exhibit complex group behavior resulting in the formation of aggregates. In other words, cockroaches prefer to be in groups and in the dark, indicating that they simultaneously optimize both the number of companions and the darkness of their shelter [18].

RIO algorithm is based on three key behaviors of cockroaches, inspired by their natural instincts. Cockroaches seek the darkest location in the search space, with darkness representing the most optimal solution based on fitness values. They also socialize with nearby cockroaches, reflecting the algorithm's use of social interactions to improve optimization. Periodically, cockroaches leave their comfort zones to explore for food, representing the algorithm's balance between exploitation and exploration, allowing agents to occasionally search for better solutions outside their current space [19].

The equations of RIO are as follows:

$$V_{i+1} = C_0 V_i + C_{max} R_1 (p_i - x_i) + C_{max} R_2 (l_i - x_i) \tag{1}$$

$$x_{i+1} = x_i + v_i \tag{2}$$

$$X_{\mathrm{i}+1} = \begin{cases} X_{\mathrm{i}} + V_{\mathrm{i}+1}, & \text{if} \left(hunger_{\mathrm{i}} < T_{\mathrm{hunger}} \right) \\ b, & \text{if } hunger_{\mathrm{i}} = T_{\mathrm{hunger}} \end{cases} \tag{3}$$

where V_i is the velocity of the agent, x_i is the current location found by agent i, p_i is the best location found by agent i, l_i is the darkest location known by agent i after socializing, C_0 is the adjustment for the momentum terms, C_{max}, R_1 and R_2. These are vectors of uniformly distributed random numbers, $hunger_i$ is an incremental hunger counter that identifies the current hunger level of a cockroach, T_{hunger} is the hunger threshold set for the cockroaches.

2.1 Modified Roach Infestation Optimization Algorithm

This method was developed by Ibidun and Adewumi it introduces a modified version of the Roach Infestation Optimization (MRIO) algorithm, which is closely tied to the social behaviors of cockroaches. MRIO improves the performance of the original Roach Infestation Optimization by incorporating partial differential equations,

crossover and mutation methods, and adding two new components: Surveillance Behavior (In this behavior, cockroaches rapidly disperse over a wider area in the search space when they sense danger. The main idea behind this model is that when the cockroach population becomes too dense in a specific area, the risk of danger increases. To avoid this risk, cockroaches need to disperse quickly.) and Cannibalism Behavior (Cannibalism is modeled as a scenario where a strong cockroach, acting as a predator, consumes a weaker cockroach, the prey, within its neighborhood. The generation of new solutions in a local neighborhood containing a predator cockroach is triggered by crossover between the two strongest local prey cockroaches, followed by mutation in the offspring prey resulting from the crossover) [18].

2.2 Genetic Roach Infestation Optimization

After introducing the Modified Roach Infestation Optimization (MRIO), which enhanced the original algorithm by incorporating surveillance and cannibalism behaviors, this section explores other significant variants of the RIO algorithm. These modifications aim to address specific challenges such as premature convergence and the balance between exploration and exploitation. The following subsections discuss two prominent adaptations: Genetic Roach Infestation Optimization (GRIO) and Dynamic Roach Infestation Optimization with Step Size Adaptation. GRIO is a novel optimization algorithm based on cockroach behavior, designed to solve continuous problems and overcome the limitations of premature convergence and local optima found in RIO. GRIO is based on key characteristics of cockroach behavior, such as seeking dark spots, socializing with friends, and reproducing with non-family members. Memory and genetic mutation facilitate adaptation. The algorithm uses a multi-swarm approach to avoid local optima, controlling cockroach speed based on the number of neighbors and the distance to the best-known solution. Additionally, GRIO introduces an adaptive, linearly decreasing inertia weight, which enhances both global and local search by adjusting velocity according to the circumstances of each particle [21].

2.3 Dynamic Roach Infestation Optimization with Step Size Adaptation

Obagbuwa et al. introduced an Euler method into the Roach Infestation Optimization (RIO) algorithm to improve swarm stability and balance exploitation with exploration. This adaptation enhances accuracy and convergence by controlling cockroach movement and preventing swarm explosion. It effectively addresses multidimensional benchmark problems up to 30 dimensions, with potential for handling more variables [20].

3 Proposed Method

In this section, we can find the flowchart of the proposed method to balance exploration and exploitation of the RIO algorithm.

Figure 1 illustrates the modified method where cockroach parameters are initialized, followed by setting up the population and calculating the distance matrix. In the first iteration, the best positions of randomly generated cockroaches are updated. If the stopping criterion isn't met, distances between neighboring cockroaches are compared; those exceeding a threshold receive random positions. A fuzzy adapter controls the C_{max} parameter to prevent algorithm explosion. Solutions are evaluated, and the hunger counter is incremented. If cockroaches exceed the hunger threshold, food is placed randomly. In the first iteration, the best positions of the randomly generated cockroaches are evaluated and updated to identify the best-performing cockroach in the population. The median of the distances (dg) between all cockroaches is also calculated. If the stopping criterion such as the maximum number of iterations or a convergence threshold is not met, the distances between neighboring cockroaches are compared. Any cockroach with a neighbor whose distance is less than the median dg will have its position reassigned randomly to avoid overcrowding.

To prevent algorithmic explosion or stagnation, a fuzzy adapter controls the C_{max} parameter, which adjusts the social and cognitive aspects of the cockroaches' behavior. As the iterations proceed, positions and velocities of the cockroaches are

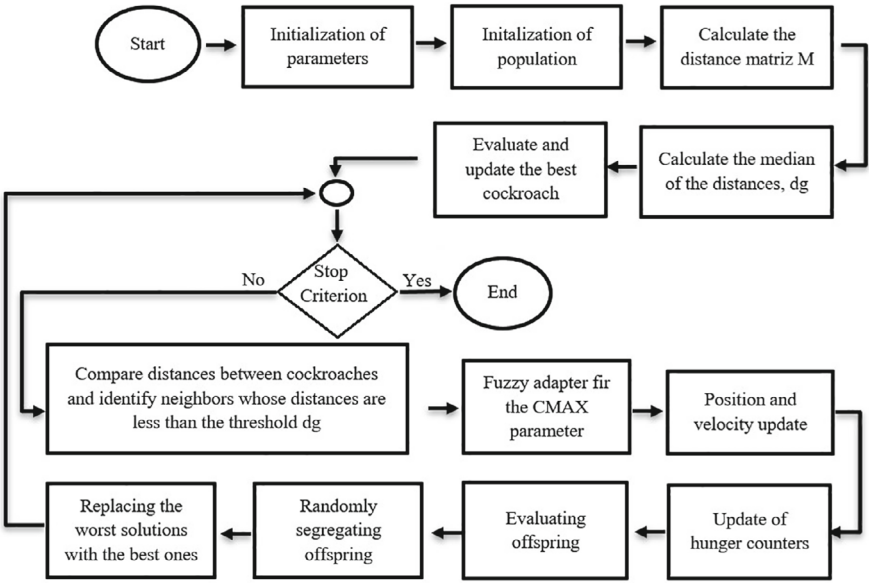

Start
Initialization of parameters
Initalization of population
Calculate the distance matriz M
Calculate the median of the distances, dg
Evaluate and update the best cockroach
Stop Criterion
No
Yes
End
Compare distances between cockroaches and identify neighbors whose distances are less than the threshold dg
Fuzzy adapter fir the CMAX parameter
Position and velocity update
Update of hunger counters
Evaluating offspring
Randomly segregating offspring
Replacing the worst solutions with the best ones

Fig. 1 Show the diagram of the method proposed as an improvement to Roach infestation optimization

updated based on this fuzzy control and interactions between individuals in the swarm.

4 Fuzzy Adaptation of RIO Parameters

In this section, we designed two fuzzy adapters to enhance the performance of the RIO algorithm. The first fuzzy adapter was constructed with iteration as the input and Cmax as the output. The second fuzzy adapter incorporates both iteration and diversity as inputs, with Cmax as the output. These adaptations aim to improve the algorithm's efficiency and robustness by dynamically adjusting the Cmax parameter based on key performance indicators.

4.1 Fuzzy Adaptation with One Input and One Output

A fuzzy adapter with one input and one output was designed, where the input is the iteration, which is used to calculate the output for the parameter $C_{max.}$

As shown in Fig. 2, a fuzzy adapter was designed to take the iteration and diversity as inputs, as these directly influence the stage in which the algorithm needs to focus on exploration or exploitation. In this way, it provides an output that affects the C_{max} parameter.

$$d = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_s} \left(x_{ij}(t) - \overline{X}_j(t)\right)^2} \tag{4}$$

Equation 4 represents the diversity.

$$f(x; a, b, c) = \begin{matrix} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ c - x, & b \le x \le c \\ \frac{c-b}{0,}, & c \le x \end{matrix} \tag{5}$$

Equation 5 represents the triangular membership functions.

In Table 1, we can see the input membership functions for our initial fuzzy adapter with 1 input and 1 output.

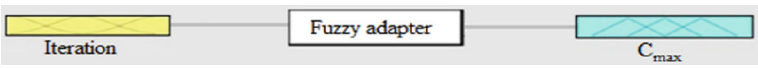


Fig. 2 Fuzzy adapter one input and one output

Table 1 Input membership functions for the one input and one output fuzzy adapter

Membership functions	a	b	b
IB (Low iteration)	0	0.20	0.35
IM (Medium iteration)	0.20	0.45	0.67
IA (Hight iteration)	0.55	0.75	1

Table 2 Output membership functions for the one input and one output fuzzy adapter

Membership functions	a	b	c
CMAXB (C_{max} low)	0.815	1	1.154
CMAXMB (C_{max} medium)	1.086	1.215	1.350
CMAXM (C_{max}hight)	1.264	1.430	1.602

In Table 2, we can find the output membership functions for our initial fuzzy adapter with 1 input and 2 output.

4.2 Fuzzy Adaptation with Two Input and One Output

A fuzzy adapter with two inputs and one output was designed, where the inputs are the iteration and diversity. These values are used to calculate the output for the parameter C_{max}.

As shown in Fig. 3 a fuzzy adapter was designed to take the iteration and diversity as inputs, as these directly influence the stage in which the algorithm needs to focus on exploration or exploitation. In this way, it provides an output that affects the C_{max} parameter.

In Table 3, we can see the input membership functions for our fuzzy adapter with two inputs and one output.

In Table 4, we can find the output membership functions for our fuzzy adapter with 2 inputs and 1 output.

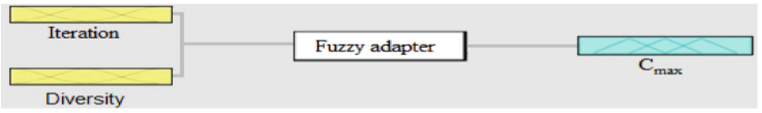


Fig. 3 Fuzzy adapter with two inputs and one output

Table 3 Input membership functions for the two inputs and one out fuzzy adapter

Membership functions	a	b	c
IB (Low iteration)	0	0.20	0.41
IM (Medium iteration)	0.25	0.50	0.75
IA (Hight iteration)	0.58	0.80	1
DB (Low diversity)	0	0.22	0.41
DM (Medium diversity)	0.25	0.50	0.70
DA (Hight diversity)	0.55	0.75	1

Table 4 Output membership functions for the two inputs and one out fuzzy adapter

Membership functions	a	b	c
CMAXB (Low C_{max})	0.915	1	1.079
CMAXMB (Medium low C_{max})	1	1.079	1.165
CMAXM (Medium C_{max})	1.079	1.165	1.251
CMAXMA (Medium hight C_{max})	1.165	1.244	1.330
CMAXA (Hight C_{max})	1.244	1.330	1.416

5 Results

In this section, benchmark mathematical functions are used to determine the efficiency of the original RIO algorithm, as well as the efficiency of the RIO algorithm with a fuzzy adapter featuring one input and one output, and the fuzzy RIO with two inputs and two outputs. The results will help us analyze how the fuzzy adaptation affects the algorithm's performance, allowing us to compare the improvements made in exploration and exploitation stages across different configurations. Furthermore, these findings will guide potential future optimizations of the fuzzy adapters.

In Table 5, we can find the results obtained from the tests with benchmark mathematical functions, in which 13 functions were used, and 2 algorithms with fuzzy parameter adaptation. The second fuzzy adaptation algorithm, which uses trapezoidal functions and a total of 9 fuzzy rules, managed to outperform the original algorithm in 8 of these functions.

The utilization of two inputs, such as iteration and diversity, proved to be highly beneficial for enhancing the performance of the algorithm. By incorporating diversity as one of the inputs, we achieved better control over the exploration process. This allows the algorithm to avoid issues like hunger explosion, which is critical for maintaining an appropriate balance between exploration and exploitation in the search space.

Moreover, when comparing these results with existing methods, it becomes evident that the dual approach provides superior adaptability to the changing environment of the problem.

Table 5 Results of benchmark functions for 10 dimensions

Functions		Original RIO	Fuzzy adapter RIO one input and one output	Fuzzy adapter RIO two inputs and one output
Sphere	$\bar{x}$	7.5703×10^{-13}	1.4516×10^{-15}	4.2799×10^{-17}
	S	4.8072×10^{-6}	4.5054×10^{-15}	2.6944×10^{-17}
Rastrigin	$\bar{x}$	7.5380×10^{0}	5.0000×10^{0}	4.9740×10^{0}
	S	4.7058×10^{0}	2.8300×10^{0}	2.4914×10^{0}
Ackley	$\bar{x}$	2.9923×10^{0}	3.0403×10^{0}	3.0084×10^{0}
	S	3.4758×10^{-2}	6.8676×10^{0}	5.6407×10^{-2}
Griewank	$\bar{x}$	5.7027×10^{-13}	5.5511×10^{-17}	6.1210^{-130}
	S	2.9085×10^{-12}	2.4868×10^{-16}	5.9915×10^{-129}
Powell	$\bar{x}$	1.1645×10^{-6}	2.9577×10^{-6}	3.9530×10^{-6}
	S	1.1877×10^{-6}	2.6258×10^{-6}	7.2045×10^{-8}
Rosenbrock	$\bar{x}$	4.9160×10^{0}	5.5960×10^{0}	5.3474×10^{0}
	S	3.7816×10^{0}	1.2203×10^{0}	1.5044×10^{0}
Alpine	$\bar{x}$	1.6604×10^{-5}	1.2448×10^{-6}	1.3316×10^{-7}
	S	1.1332×10^{-3}	6.7529×10^{-6}	5.5942×10^{-7}
Zakharov	$\bar{x}$	5.5627×10^{-1}	4.7523×10^{-5}	3.6596×10^{-6}
	S	7.2714×10^{-1}	1.2903×10^{-4}	1.0570×10^{-5}
Sum squares	$\bar{x}$	2.9145×10^{-11}	2.2171×10^{-15}	7.6189×10^{-16}
	S	1.1692×10^{-10}	9.0514×10^{-15}	3.8907×10^{-15}
Levy	$\bar{x}$	1.7985×10^{-13}	5.1043×10^{-15}	1.8371×10^{-18}
	S	9.4055×10^{-13}	1.8415×10^{-15}	4.3459×10^{-18}
Dixon price	$\bar{x}$	3.6792×10^{-1}	3.4688×10^{-1}	2.1104×10^{-1}
	S	3.3081×10^{-1}	3.4688×10^{-1}	3.3301×10^{-1}
Wood	$\bar{x}$	2.0433×10^{-2}	2.5556×10^{-2}	2.9631×10^{-2}
	S	3.2841×10^{-2}	3.7777×10^{-2}	3.8524×10^{-2}
Dejong1	$\bar{x}$	2.8678×10^{-15}	8.6483×10^{-15}	3.5753×10^{-15}
	S	5.2414×10^{-15}	4.5313×10^{-15}	1.9556×10^{-14}

Furthermore, when comparing these results with existing methods, it is evident that the dual approach provides superior adaptability to the changing environment of the problem. While previous algorithms such as PSO (Particle Swarm Optimization) or GA (Genetic Algorithm) may have suffered from premature convergence, our method, by including diversity as an input, allows for broader and more effective exploration, reducing the risk of stagnation in local optima.

6 Conclusions

The algorithm is very good at exploration, but this limits its performance in exploitation, leading to inefficient convergence. Therefore, the algorithm needs to balance exploration and exploitation. With the help of a fuzzy adapter, the C_{max} parameter was controlled, allowing more efficient management of both exploration and exploitation, thus improving the overall efficiency of RIO. By fine-tuning this balance, the fuzzy adapter enhances the algorithm's ability to find optimal solutions more quickly, reducing convergence time and increasing performance across various problem domains.

References

1. F. Valdez, J. C. Vazquez, and F. Gaxiola, "Fuzzy Dynamic Parameter Adaptation in ACO and PSO for Designing Fuzzy Controllers: The Cases of Water Level and Temperature Control," Advances in Fuzzy Systems, vol. 2018, 2018, https://doi.org/10.1155/2018/1274969.
2. D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," Soft Computing, vol. 22, no. 2, pp. 387–408, 2017, https://doi.org/10.1007/s00500-016-2474-6.
3. P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, "Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic," Expert Systems with Applications, vol. 40, no. 8, pp. 3196–3206, 2013, https://doi.org/10.1016/j.eswa.2012.12.033.
4. P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, "Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic," Expert Systems with Applications, vol. 40, no. 8, pp. 3196–3206, Jun. 2013, https://doi.org/10.1016/j.eswa.2012.12.033.
5. M. Guerrero, F. Valdez, and O. Castillo, "Comparative Study between Type-1 and Interval Type-2 Fuzzy Systems in Parameter Adaptation for the Cuckoo Search Algorithm," Symmetry (Basel), vol. 14, no. 11, Nov. 2022, https://doi.org/10.3390/sym14112289.
6. S. A. Al-Agamy, F. Mutaher Ba-Alwi, and A. M. Mohsen, "A Fuzzy Logic for Parameter Adaptation in Ant Colony Optimization Approach," 2022. [Online]. Available: www.ijisrt.com.
7. S. Sennan, S. Ramasubbareddy, S. Balasubramaniyam, A. Nayyar, M. Abouhawwash, and N. A. Hikal, "T2FL-PSO: Type-2 Fuzzy Logic-Based Particle Swarm Optimization Algorithm Used to Maximize the Lifetime of Internet of Things," IEEE Access, vol. 9, pp. 63966–63979, 2021, https://doi.org/10.1109/ACCESS.2021.3069455.
8. F. Olivas, F. Valdez, O. Castillo, and P. Melin, "Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic," Soft Computing, vol. 20, no. 3, pp. 1057–1070, Mar. 2016, https://doi.org/10.1007/s00500-014-1567-3.
9. S. Sharmila and S. Vijayarani, "Association rule mining using fuzzy logic and whale optimization algorithm," Soft Computing, vol. 25, no. 2, pp. 1431–1446, Jan. 2021, https://doi.org/10.1007/s00500-020-05229-4.
10. R. Rajabioun, "Cuckoo Optimization Algorithm," Applied Soft Computing, vol. 11, no. 8, pp. 5508–5518, 2011, https://doi.org/10.1016/j.asoc.2011.05.008.
11. L. Amador-Angulo and O. Castillo, "An Interval Type-2 Fuzzy Logic Approach for Dynamic Parameter Adaptation in a Whale Optimization Algorithm Applied to Mathematical Functions," Axioms, vol. 13, no. 1, p. 33, Dec. 2023, https://doi.org/10.3390/axioms13010033.

12. I. Miramontes and P. Melin, "Enhancing Dynamic Parameter Adaptation in the Bird Swarm Algorithm Using General Type-2 Fuzzy Analysis and Mathematical Functions," Axioms, vol. 12, no. 9, Sep. 2023, https://doi.org/10.3390/axioms12090834.
13. S. Mirjalili, J. Song Dong, A. S. Sadiq, and H. Faris, "Genetic algorithm: Theory, literature review, and application in image reconstruction," in Studies in Computational Intelligence, vol. 811, Springer Verlag, 2020, pp. 69–85, https://doi.org/10.1007/978-3-030-12127-3_5.
14. D. Hidalgo, L. Cervantes, O. Castillo, P. Melin, and R. M. Soto, "Fuzzy parameter adaptation in genetic algorithms for the optimization of fuzzy integrators in modular neural networks for multimodal biometry," Computación y Sistemas, vol. 24, no. 3, pp. 1093–1105, 2020, https://doi.org/10.13053/CYS-24-3-3329.
15. F. Valdez, O. Castillo, and C. Peraza, "Fuzzy Logic in Dynamic Parameter Adaptation of Harmony Search Optimization for Benchmark Functions and Fuzzy Controllers," International Journal of Fuzzy Systems, vol. 22, no. 4, pp. 1198–1211, Jun. 2020, https://doi.org/10.1007/s40815-020-00860-7.
16. D. Yahya Qassem and D. Akram Al, "A Solution to the Next Release Problem by Swarm Intelligence." [Online]. Available: www.techniumscience.com.
17. H. C. Tsai, "Roach infestation optimization with friendship centers," Engineering Applications of Artificial Intelligence, vol. 39, pp. 109–119, Mar. 2015, https://doi.org/10.1016/j.engappai.2014.12.003.
18. O. Ibidun and A. Adewumi, "A Modified Roach Infestation Optimization", School Of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, Durban, 4000.
19. T. Havens, C. Spain, N. Salmon, and J. Keller, "Roach Infestation Optimization," 2008 IEEE Swarm Intelligence Symposium, 2008, https://doi.org/10.1109/sis.2008.4668317.
20. I. C. Obagbuwa, A. Adewumi, and A. Adebiyi, "A dynamic step-size adaptation roach infestation optimization," 2014 IEEE International Advance Computing Conference (IACC), https://doi.org/10.1109/iadcc.2014.6779498.
21. A. Gustavo and L. Brites, "Genetic Roach Infestation Optimization-Optimization Algorithm based on Cockroaches' social life".
22. L. Amador-Angulo and O. Castillo, "An Interval Type-2 Fuzzy Logic Approach for Dynamic Parameter Adaptation in a Whale Optimization Algorithm Applied to Mathematical Functions," Axioms, vol. 13, no. 1, p. 33, 2023, https://doi.org/10.3390/axioms13010033.
23. H. Carreon-Ortiz, F. Valdez, and O. Castillo, "Comparative Study of Type-1 and Interval Type-2 Fuzzy Logic Systems in Parameter Adaptation for the Fuzzy Discrete Mycorrhiza Optimization Algorithm," Mathematics, vol. 11, no. 11, 2023, https://doi.org/10.3390/math11112501.
24. M. L. Lagunes, I. Miramontes, O. Castillo, and P. Melin, "Comparative Study of Gorilla Troops Optimizer and Stochastic Fractal Search with fuzzy dynamic parameter adaptation," Computación y Sistemas, vol. 28, no. 2, 2024, https://doi.org/10.13053/cys-28-2-5028.

From Containers to the Cloud, Scaling Event-Driven Evolutionary Algorithms

Mario Garcia-Valdez, Alejandra Mancilla, and Juan J. Merelo Guervós

Abstract This paper presents the design choices, challenges, and practical issues of scaling evolutionary algorithms (EAs) in the cloud using containers. To exemplify these challenges practically, we present a case study where we evolve a fuzzy controller using an event-driven EA. We show how EAs can be scaled by using multiple containers in parallel to speed up the evolutionary process. We present some preliminary results on the performance of the event-driven EA when running locally and in the cloud.

Keywords Cloud computing · Event-driven architectures · Evolutionary algorithms

1 Introduction

Cloud computing allows individuals and organizations to access services such as virtual machines, storage, networking, and software, paying only for the resources they use. Many providers offer free tiers, reducing the cost of adoption. Cloud resources and their configurations are often defined through scripts and text-based setups, making them easily reproducible by others. These benefits have made cloud computing not only a standard in enterprise software development but also a growing trend in computer science for running computational experiments [1, 2]. Typically,

M. Garcia-Valdez (✉) · A. Mancilla
Department of Graduate Studies, Tecnológico Nacional de México,
Tijuana, Baja California, Mexico
e-mail: mario@tectijuana.edu.mx

A. Mancilla
e-mail: alejandra.mancilla@tectijuana.edu.mx

J. J. Merelo Guervós
Department of Computer Engineering, Automatics and Robotics, University of Granada,
Granada, Spain
e-mail: jmerelo@ugr.es

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_22

these experiments are developed on personal computers, workstations, or low-cost virtual machines and then scaled to parallel execution using multiple instances to increase speed. Data from experiments, such as logs or distributed storage results, can be collected and shared alongside the code and configuration used, enabling full transparency and reproducibility.

Successful examples of integrating cloud resources and scientific applications are web-based notebooks [3], which combine text and executable code in a single document. The code running in notebooks can run on cloud resources and seamlessly scale using more computing resources, including high-end GPUs. For instance, Google Colaboratory [4] offers a cloud-based environment for running Jupyter Notebooks, allowing users to scale to meet their computational needs. A significant advantage for scientific applications is that the code can be easily shared and reproduced by other researchers. The specifications of the computing resources can also be shared easily.

Additionally, numerous standards, open specifications, and open software allow scientists and students to choose between cloud providers or even run experiments across multiple platforms.

The problem is that some experiments are not as straightforward as a single-page script. For instance, distributed applications require several node components interacting in parallel through a network. Sharing and scaling this code is more complex, but this is precisely the type of problem that cloud computing technologies are trying to solve. Before we discuss the sharing of specifications, we will cover the main components of a design pattern based on microservice architectures [5] and serverless computing principles [6]. Microservices [1] break down applications into smaller, loosely coupled components, improving scalability and flexibility. Serverless computing allows users to upload just the code (such as lambda functions) without worrying about managing the underlying infrastructure, which dynamically adapts to the demands. Combined, these patterns improve the efficiency and adaptability of computational experiments in cloud environments.

Microservices and Serverless functions are typically executed in Docker containers or similar technologies. Containers encapsulate the runtime environment, providing all the resources the service needs to run independently, including the Linux operating system distribution, binaries, libraries, programming languages, runtime environments, and anything else required to operate. This configuration is specified in a particular file called `Dockerfile`, which creates an Image from which several container instances can be launched. Even more, a service usually consists of several containers that interact with each other. In Docker's case, this is specified in another configuration file called `docker-compose.yml`.

With these specifications, users can execute the 'service,' in our case, a computational experiment on their local computers, or share these specifications with other scientists to validate the results. An interesting use case is to upload these specifications to a Cloud Service to execute the experiment using scalable resources.

In this paper, we present a use case in which we developed an evolutionary algorithm using a Microservice architecture using Docker containers and then scaled the local implementation by deploying the container solution to Amazon's Elastic

Container Service (Amazon ECS). Our focus is on the practical issues found in these kinds of implementations.

This work presents a case study demonstrating the application of an event-driven architecture, implemented using containers and a lightweight orchestration tool. Specifically, we apply this architecture to scale an evolutionary algorithm to evolve a fuzzy controller. Additionally, we compare local and cloud deployments, evaluating both speed-up and cost considerations.

We can highlight the particular contributions of this work as a case study of the application of an event-driven architecture implemented with containers and a basic orchestration tool. Furthermore, using the architecture, we scale an evolutionary algorithm to evolve a fuzzy controller. A comparison between a local and cloud deployment considering the speed-up and cost is also presented.

We organized the paper as follows: First, Sect. 2 presents the background, related works, and the proposed method. Section 3 presents the use case, while Sect. 4 presents the container-based cloud deployment and experimental setup, discussing the results in Sect. 5. Finally, we offer the conclusions and suggestions for future work in Sect. 6.

2 Background and Related Work

We utilized the Genetic Algorithm (GA) proposed by Holland [7], a population-based metaheuristic algorithm used primarily for optimization. In a GA, a random population of candidate solutions is improved using mechanisms inspired by biological evolution. First, a population of random candidate solutions is generated. Once we have a random population of candidate solutions, we assess the quality of each solution using a fitness function. Candidate solutions with higher fitness have a higher probability of survival. This concept is used in the next step, where candidate solutions are selected for reproduction to breed the next generation of the population. Breeding involves the *crossover* and *mutation* operators. The crossover genetic operator combines genetic information from both parents to produce new offspring. With some probability, the newly generated candidate solutions go through the mutation operator; this genetic operator adds diversity to the population by changing the values of some of the genes of a candidate solution. The population evolves for several generations until an optimal or suitable solution is found.

The most costly task of GAs is the evaluation of the candidate solutions, and since the beginning, developers have tried to run fitness functions in parallel. A common approach is to distribute among several computers or processors just the evaluation of the fitness functions, leaving all the other components of the algorithm to a single primary node. Another common approach is to distribute copies of the GA to several nodes in what is called a multi-population or islands model [8, 9]. In a multi-population algorithm, each population runs in isolation for several generations, but after some time, we perform a migration step in which some candidate

solutions are exchanged between populations. Migration is essential to prevent premature convergence of the candidate solutions to a local minimum [10]. There is also some evidence that multi-population algorithms can outperform single-population implementations.

In this work, we use a multi-population genetic algorithm in which islands are executed in isolated docker containers using an event-driven architecture. An event-driven system comprises two main components: Events and Event Handlers. Events can be user actions, sensor readings, or messages. On the other hand, event handlers are functions that react to events as they happen. Event-driven systems are asynchronous because there is a constant stream of events, and event handlers do not block the flow of events. A central component of this architecture is an event loop or message broker, which continuously receives events and dispatches them to the corresponding handler. Furthermore, there is a decoupling between the event generation and the handling of events. This means we can have many copies of a handler without affecting the producers of events. The event-driven architecture we use in this paper follows the EvoSwarm model described in detail in a previous work [11]. To implement the GA as an event-driven system, we have the following components:

- Events: In this case, candidate solution populations are the messages received by event handlers. Populations are encoded as text in a `json` format and have additional metadata indicating the parameters for executing the algorithm on the population.
- Message Queues: These are responsible for receiving and delivering the messages to the appropriate event handlers. In this case, handlers are responsible for constantly taking messages from their corresponding message queue once they are available to do more work.
- Event Handlers: Event handlers are methods responsible for taking population messages and running a GA over them using the parameters in the message. Once a population is processed, the modified population is returned to another message queue and takes another message.
- Combinator: This is a particular node responsible for keeping track of the evolutionary algorithm. It generates the initial population messages and takes the results to verify if the optimal solution has been found and to migrate individuals between populations. Keeps the algorithm's loop running by sending new populations to the message queue.

We can see the main components of the EvoSwarm architecture in Fig. 1.

As we mentioned earlier, we can implement this event-driven architecture using docker containers for each of the components of the system. Next, we need to deploy these components to the Cloud. We can consider several options to do this:

- We can use a Kubernetes cluster; this deployment type provides auto-scaling capabilities and other capabilities to run volatile, high-demand, mission-critical applications. This makes the configuration of a Kubernetes cluster a more difficult task. This option has been used by Zhao et al. [12] and Dziurzanski et al. [13] to run multi-population algorithms.

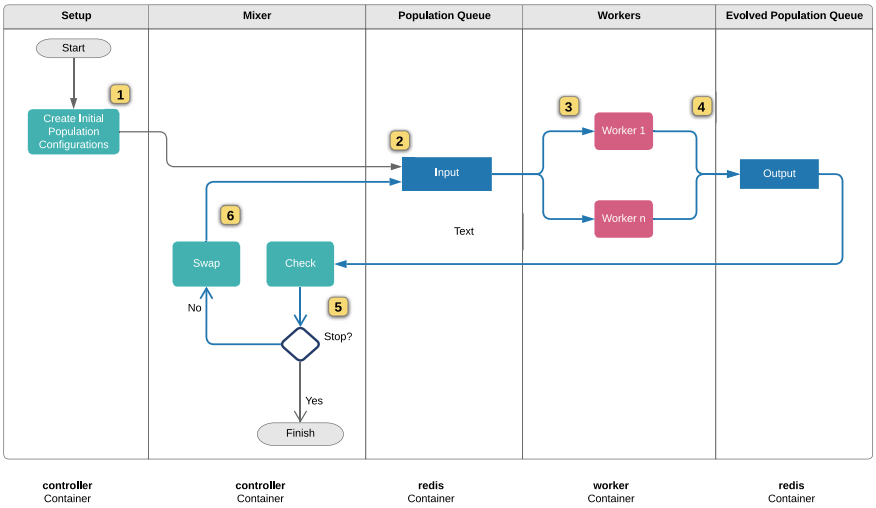


Fig. 1 Event-driven architecture we use in this paper, the figure shows the containers in which each process is run as swimlanes

- Another option is to use a Container Orchestration service. These services, like Amazon Elastic Container Service (ECS) or Red Hat OpenShift, are `docker-compose` compatible and can provide transparent provisioning. Several works are using this type of service [14].

Because Kubernetes adds more complexity and some of the capabilities are unnecessary for a single-run experiment in which the demand for resources remains constant during the life of the experiment, and high availability is not a requirement, it is that we use AWS ECS in this work.

3 Case Study: Evolving a Fuzzy Controller

As a use case, we will evolve the parameters of a Fuzzy Controller that keeps a two-wheel robot in a path. This is a case in which the cost of evaluating each candidate solution is high because to test the parameters, each controller needs to be tested by running several simulations involving numerical libraries for solving Ordinary Differential Equations (ODE) and optimization methods to find the shortest distance to the path. We use the GA to optimize 15 continuous parameters of the membership functions defining the fuzzy controller. We create a controller instance and simulate its performance across several paths. We then calculate the fitness of the controller using the average RMSE (Root Mean Square Error) for each path. The simulation lasts 50 units of time ($t = 50$), but if a controller cannot adequately follow a path (the error exceeds an RMSE of 100 at any given time, the simulation is terminated early,

Table 1 Parameters used for the GA algorithm running in each population. Random values are indicated by a range

Parameter	Value
Crossover probability	[0.3, 0.9]
Crossover method	Single point
Selection method	Tournament $k = 3$
Mutation probability	[0.1, 0.5]
Mutation type	Gaussian
List size	15
Function evaluations	2500

and the controller is assigned the worst possible fitness value (RMSE = 5000). Also, if after the simulation ends with the robot far from the target, an RMSE of 2000 is assigned as the fitness. This means that the time needed to run a simulation is not constant.

The GA and the path-tracking simulator are implemented in Python, meaning that despite the availability of multiple CPUs, the Python interpreter only utilizes a single thread or CPU at any given time. For this reason, our distributed implementation uses only single CPU containers to handle the work of each population. The parameters of the GA are shown in Table 1. Specific parameters are chosen randomly to balance exploitation and exploration by having populations more inclined to use one of the two search strategies. This means that at the beginning of the run, some populations will explore, and at the end, some populations will exploit.

4 Deployment

As we mentioned earlier, EC2 orchestrates container-based applications. There are two ways of launching an application in EC2. The first is using classical EC2 instances and infrastructure. This gives the developer the freedom to choose and highly tune each instance to the problem at hand, at the cost of manually managing the EC2 infrastructure. The other option, Fargate, automatically manages EC2 instances, and the user only needs to specify the resources needed for each container, similar to a docker-compose file. EC2 uses the concept of a Task to specify the resources, environment variables, networking, and permissions used by one or more containers. There is also the concept of a Service, in which several Tasks implement a service that needs to run continuously, with auto-scaling and load balancing. Services are more suited to implementing web services or other internet applications that need to be always present. In our case, we do not need this kind of persistence because our experiments are one-time events. An important AWS service needed is monitoring and logging, which, in this case, is provided automatically by the CloudWatch service.

Furthermore, to execute a service (in our case, an experiment), we must create an ECS Cluster. Clusters are logical groups of EC2 instances or Fargate Tasks. Finally, we can use Amazon's container registry to keep track of docker images.

4.1 Local Deployment

As a baseline, we implemented the containerized implementation described earlier and ran the experiment on a Workstation computer, in this case, a 2022 MacStudio computer, with the M1 Ultra CPU, 128 GB of RAM, and macOS Sonoma 14.6.1. The M1 Ultra consists of two M1 Max units connected with UltraFusion Interconnect with 20 CPU cores and 96 MB system-level cache (SLC). We used the Docker Engine v20.10.21, with the VirtioFS accelerated directory sharing enabled, version 4.14 flags. We run the Redis and Worker containers with the `-scale` parameter set to eight. This means that eight instances of the worker containers were launched. The controller component was executed in another terminal as a script. It is important to note that the M1 Ultra CPU uses an ARM architecture that needs a virtualization layer to run Docker. This is also true for Windows machines because Docker is a native Linux application. Nevertheless, the virtualization layer is highly optimized. The `docker-compose` file is as follows:

```
version: '3'

services:
  redis:
    image: "redis:alpine"
    ports:
    - "6379:6379"
  worker:
    build: '.'
    depends_on:
      - redis
    command: python worker.py
    environment:
      PYTHONUNBUFFERED: 1
      REDIS_HOST: redis
```

Listing 1 `docker-compose` used for deploying the containers.

There are two environment variables to specify: `PYTHONUNBUFFERED` enables the output of print commands used for logging results and other details, and the `REDIS_HOST` used for specifying the IP address of the Redis container. In a local setting, the IP address is unnecessary because containers can access each other by name. In this case, `redis` is the container's name. However, this variable is essential when the containerized application is deployed to the cloud. When the containers are launched, the worker script is started with the command: `python worker.py`.

Table 2 Parameters used for the distributed multi-population GA algorithm

Parameter	Value
Population size	10
Generations	5
Number of populations	10
Number of cycles	5
Function evaluations	2500

Each worker is launched using the following `Dockerfile`:

```
FROM python:3.7.8

COPY . /Worker
WORKDIR /Worker
RUN pip install -r requirements.txt
RUN chmod 644 worker.py
```

Listing 2 `Dockerfile` used for deploying Worker containers.

This script copies the folder with the code to the Worker directory inside the container, installs the necessary libraries, and makes the script executable. Table 2 details the parameters of the locally distributed experiment. The work is distributed among several containers. Each of these runs on a separate CPU or thread, and the communication between them uses the Docker engine's virtual network. This also has the advantage of avoiding a limitation of the Python interpreter that limits the execution of Python byte code to one thread at a time. In this case, we have several copies of Python code, running each in a different container. The global population is divided into smaller populations, and in this case, there are ten candidate solutions for each population. Also, five generations may seem small, but we must consider that generations are stopped when one worker finishes, but they continue in another when the population is taken by another worker. The number of cycles is the number of times a population is processed by a worker. If we multiply the parameters (generations, cycle, population size, and number of populations), we get the number of function evaluations of the algorithm, which, in this case, is 2500.

We consider the number of function evaluations to be the work needed to complete the GA. This will remain more or less constant in the experiments. The variable we want to minimize is the time needed to complete these evaluations.

4.2 AWS Deployment

We tested both options for containerized deployment in AWS: Fargate and EC2. For both deployment types, we must upload the container images to a registry service and create Tasks indicating the type of deployment and container details. We can scale the number of containers in a Task.

Fargate is the more practical deployment because the resources needed to execute containers are provided automatically. Specify the number of containers, and they will be executed instantly. On the other hand, with the EC2 deployment, we must first launch a certain number of EC2 instances in which the containers will be deployed. The advantage is that we have more control over the type of instance we choose, and we can even enable `ssh` to connect with the instances. When launched, containers are automatically assigned to one of the available instances. Users are responsible for providing instances with enough CPU, memory, and network cards to execute the number of Tasks we need.

We tested various types of EC2 configurations with different types and number of instances. First, we tested the same configuration as the local machine and then increased the total number of vCPUs, looking for faster times than the local implementation. When we increase the number of Workers, we also need to increase the number of populations. We changed the distributed parameters to keep the overall number of function evaluations the same as the local implementation.

The EC2 options are detailed next.

We tested using instances that are suited for high-end processing and have a higher cost. To run the number of containers needed, we also need several EC2 instances; although some can run the experiment locally, they have limitations when used in an ECS cluster. For instance, each container is assigned a virtual network card and IP address, and some instances have a limit on the number of virtual cards available. Also, there is a limitation on the number of CPUs and RAM available. The number of instances is shown in parentheses in Table 3 along with the price by the hour and the distributed configuration. It is also essential to clarify that the price reported is the cost per CPU hour. This means that for the most expensive instance, running 22 workers (each in a CPU), the hour costs around 72 USD.

5 Results

Table 4 shows the average ($n = 15$) number of seconds needed to run an experiment in each configuration. First, running the experiment locally is faster than most cloud configurations. We need three high-cost instances to achieve similar and better results. In the case of Fargate, although it is straightforward to launch and operate, the CPU power is also limited. It is a good match for running operations that are not computationally demanding. In this case, it takes almost three times more time to complete the experiments with the Fargate configuration. The mid-range options

Table 3 EC2 instance types used in the experiments. The number of instances is shown in parentheses. The price is the average reported by https://aws-pricing.com/ at the time of writing. Fargate CPU is not specified, and it is selected from currently available resources, it can be an Intel Xeon or AMD EPYC class CPU

EC2 instance	CPU	Price	Wks	Pops	PSize	Gens	Cycles
Fargate	Not available	0.4048	8	10	5	5	10
c7a.2xlarge (3)	AMD EPYC 9R14	0.4451	8	10	5	5	10
z1d.3xlarge (3)	Intel Xeon Platinum 8151	1.2608	8	10	5	5	10
r7iz.8xlarge (2)	Intel Xeon Scalable (Sapphire Rapids)	3.248	8	10	5	5	10
r7iz.8xlarge (3)	Intel Xeon Scalable (Sapphire Rapids)	3.248	16	20	5	5	5
r7iz.8xlarge (3)	Intel Xeon Scalable (Sapphire Rapids)	3.248	22	25	5	4	5

Table 4 EC2 instance types used in the experiments. The price is the average reported by https://aws-pricing.com/ at the time of writing

Computing resource	Time in seconds	Evaluations per second
MacStudio	247.50	10.101
Fargate	713.38	3.504
c7a.2xlarge (3)	471.60	5.301
z1d.3xlarge (3)	467.53	5.347
r7iz.8xlarge (2)	330.28	7.569
r7iz.8xlarge (3)	160.22	15.603
r7iz.8xlarge (3)	129.29	19.3353

z1d.3xlarge and c7a.2xlarge are faster than the Fargate option, and this means that they are a viable alternative, but they do not offer a clear advantage in computing time. We need to move to the more expensive alternative to have faster execution times. When using the r7iz.8xlarge solution with three instances and launching 22 containers, we see faster times, achieving almost twice the evaluations per second than the local implementation. This means we can effectively reduce the time to find a solution, but we must use more resources.

6 Conclusion

There is a cost added to the management of the computing resources offered by the cloud provider, as well as the added high availability requirements. These are not always needed for these kinds of experiments. In a worst-case scenario, we only need

to run the experiment again and collect the data. Of course, there are some instances in which this extra cost is justified. Containers can significantly simplify the local management of computing resources already available in the local lab, with easy reproducibility to other computing environments.

In future work, we will continue with a more comprehensive comparison with other orchestration tools, in particular, Docker Swarm, because it follows a minimalistic approach and can be easily implemented in a LAN. Furthermore, we must also consider the evolution of other technologies, in particular, quantum computing, which is a promising computing resource that is already available as a cloud service.

Acknowledgements This research was funded by TecNM grant number 21094.24-P, the Spanish Ministry of Science, Innovation and Universities MICIU/AEI/10.13039/501100011033 under project number PID2023-147409NB-C21, and DemocratAI PID2020-115570GB-C22.

References

1. Malawski, M., Gajek, A., Zima, A., Balis, B., Figiela, K.: Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions. Future Generation Computer Systems **110**, 502–514 (2020) https://doi.org/10.1016/j.future.2017.10.029
2. Scholl, B., Swanson, T., Jausovec, P.: Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications. " O'Reilly Media, Inc.", ??? (2019)
3. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B.E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J.B., Grout, J., Corlay, S., *et al.*: Jupyter notebooks-a publishing format for reproducible computational workflows. Elpub **2016**, 87–90 (2016)
4. Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V.H.C., Reboucas Filho, P.P.: Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access **6**, 61677–61685 (2018)
5. Thönes, J.: Microservices. IEEE Software **32**(1), 116–116 (2015) https://doi.org/10.1109/MS.2015.11
6. Varghese, B., Buyya, R.: Next generation cloud computing: New trends and research directions. Future Generation Computer Systems **79**, 849–861 (2018)
7. Holland, J.H.: Genetic algorithms. Scientific american **267**(1), 66–73 (1992)
8. Gong, Y., Fukunaga, A.: Distributed island-model genetic algorithms using heterogeneous parameter settings. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 820–827 (2011). IEEE
9. Araujo, L., Merelo Guervós, J.J., Cotta, C., Vega, F.F.: Multikulti algorithm: Migrating the most different genotypes in an island model. arXiv preprint arXiv:0806.2843 (2008)
10. Li, Y., Zeng, X.: Multi-population co-genetic algorithm with double chain-like agents structure for parallel global numerical optimization. Applied intelligence **32**(3), 292–310 (2010)
11. García-Valdez, M., Merelo, J.J.: Event-driven multi-algorithm optimization: Mixing swarm and evolutionary strategies. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pp. 747–762 (2021). Springer
12. Zhao, S., Dziurzanski, P., Przewozniczek, M., Komarnicki, M., Indrusiak, L.S.: Cloud-based dynamic distributed optimisation of integrated process planning and scheduling in smart factories. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1381–1389 (2019)

13. Dziurzanski, P., Zhao, S., Przewozniczek, M., Komarnicki, M., Indrusiak, L.S.: Scalable distributed evolutionary algorithm orchestration using docker containers. Journal of Computational Science **40**, 101069 (2020)
14. Salza, P., Ferrucci, F.: Speed up genetic algorithms in the cloud using software containers. Future Gener. Comput. Syst. **92**, 276–289 (2019) https://doi.org/10.1016/j.future.2018.09.066

Comparative Evaluation of Machine Learning Algorithms for Predicting Defects Using Object-Oriented and Software Complexity Metrics

Yaretxy Arely Pelayo Lomeli, Juan Pablo García Vázquez, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Jesus Eduardo Soto Vega, and Araceli Celina Justo López

Abstract Software metrics play a key role in predicting software defects. Various types of metrics have been proposed, such as software complexity and object-oriented metrics. Considering the above, this chapter presents a comparative analysis of several traditional machine learning algorithms, such as K Nearest Neighbors (KNN), Vector Support Machines (SVM), Random Forest (RF), and a multilayer perceptron (MLP). They were trained with various software metrics. Three experiments were conducted. In the first, the algorithms were trained with software complexity metrics. In the second, object-oriented metrics were used, and finally, in the third, the combination of software complexity and object-oriented metrics was used. Our results demonstrate that using the combination of object-oriented and complexity software metrics allows defects to be predicted more accurately than using a single type of metric. The algorithms with the best performance for each experiment, respectively, were KNN with an accuracy of 80.26% and F1-Score of 0.8332 and G-Mean of

J.P. García Vázquez and M.A. Astorga-Vargas—Equal contribution.

Y. A. Pelayo Lomeli · J. P. García Vázquez (✉) · M. A. Astorga-Vargas · J. E. Ibarra Esquer · J. E. Soto Vega · A. C. Justo López
Facultad de Ingeniería, MyDCI, Universidad Autónoma de Baja California (UABC), Mexicali, Baja California, Mexico
e-mail: pablo.garcia@uabc.edu.mx

Y. A. Pelayo Lomeli
e-mail: yaretxy.pelayo@uabc.edu.mx

M. A. Astorga-Vargas
e-mail: angelicaastorga@uabc.edu.mx

J. E. Ibarra Esquer
e-mail: jorge.ibarra@uabc.edu.mx

J. E. Soto Vega
e-mail: soto.jesus@uabc.edu.mx

A. C. Justo López
e-mail: araceli.justo@uabc.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_23

0.7994, KNN with an accuracy of 80.76% and F1-Score of 0.8309 and G-Mean of 0.8013 and finally, RF with an accuracy of 85.52% and F1-Score of 0.8667 and G-Mean of 0.8726.

Keywords Machine learning · Defect prediction · Software testing

1 Introduction

Currently, the object-oriented programming paradigm is the most widely used in software development [1]. According to the IEEE's annual report The top programming lenguages, the most widely used programming languages in software development are object-oriented languages in which Python, Java, C++, C#, and PHP stand out [2]. Despite the existence of methodologies for software development under the OOP paradigm, software continues to have defects today [3].

To ensure software quality, one of the key aspects is to prevent defects in the software, this through software testing, which allows identifying problems in a program that trigger an undesired result, such as crashes or data loss [4]. Software testing is a fundamental activity within the software development process [3, 5]. For the design and execution of tests, ISO/IEEE 29119:2022 suggests three techniques: (1) Tests based on specifications associated with functional requirements, these tests are also known as black box tests because they are performed at the level of user interfaces without evaluating the internal structure of the software, (2) Tests based on the internal structure in which the code is explored, these tests are known as white box tests, and (3) Tests based on experience by the human evaluator. Black box tests are the most applied but are less effective in detecting errors [6], compared to white box tests, because their design is more complex as they have to ensure that paths, instructions and branches are used during the execution of the test. This makes software testing a complex and expensive activity, especially when the software becomes more complex [3, 7]. Therefore, it is not possible to test, identify, and correct software defects completely [3, 5]. In this sense, software metrics have been established for defect forecasting, based on different software attributes such as size, complexity, coupling, among others [1, 7] In particular, object-oriented software metrics play a key role in the prediction of defects [1]. On the other hand, to optimize testing, automated tools have been developed aimed at functional and non-functional testing. Examples of these tools are Unified Functional Testing Microfocus [8], Selenium [9], Eggplant [10], Ranorex [11], Test complete [12], Microsoft Visual Tests [13], among others.

It is observed that derived from the potential of Machine Learning to process large volumes of data, interest has arisen from both practitioners and researchers to improve the effectiveness and efficiency of tests by applying different algorithms and techniques for defect prediction [3, 14, 15]. Machine learning, defined as "a branch of artificial intelligence that allows computers to learn from data and make predictions or decisions based on that data, without being explicitly programmed to do so" [16], has been considered as an alternative for training models with character-

istics associated with software quality (e.g., object-oriented metrics) for the purpose of identifying patterns or indicators that can help predict the occurrence of software defects [17, 18]. Considering the above, this chapter presents a comparative analysis of several classical machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), and a multilayer perceptron (MLP). These algorithms were trained using various software metrics to predict software defects. Three experiments were conducted: (1) the algorithms were trained with software complexity metrics; (2) the algorithms were trained with object-oriented metrics; and finally, (3) the algorithms were trained with a combination of software complexity and object-oriented metrics.

This chapter is organized as follows. In Sect. 2, several studies are presented that report the use of object-oriented metrics and their combination with other software metrics to predict defects. The dataset used, the machine learning algorithms implemented and their hyperparameters, data preprocessing techniques, performance evaluation metrics and a description of the experimentation environment are presented in Sect. 3. The results of the experiments carried out are presented in Sect. 4. A discussion of the results is presented in Sect. 5. Finally, the results and future work are discussed in Sect. 6.

2 Related Work

In the literature, various object-oriented metrics have been presented to measure structural properties of object-oriented software [19]. The combination of object-oriented metrics with other software metrics, such as McCabe and Halstead software complexity, centrality measures [20], to name a few, have been considered to generate models for predicting software defects. For example, Ouellet and Badri [21] present the combination of object-oriented metrics and centrality measures. This is in order to investigate whether centrality measures improve quality from the perspective of fault-proneness prediction, fault severity and number of faults. To do this, they classified object-oriented metrics into three categories: (1) Complexity: Average Cyclomatic Complexity (ACC), Response For a Class (RFC), and Weighted Method Complexity (WMC); (2) Coupling: Afferent Coupling (CA), Coupling Between Objects (CBO), and Efferent Coupling (CE); and (3) Size: Class Interface Size (CIS), Source Lines of Code (LOC), and Number of Methods (NOM). Whereas, the centrality measures used were Betweenness Centrality (BC), Closeness Centrality (CL), Density of Maximum Neighborhood Component (DMNC), Eigenvector Centrality (EV), Katz Centrality (KATZ), Leverage Centrality (LE), Lobby Centrality (LO), Semi-local Centrality (SLC), and Vertex Entropy Centrality (VEC). The metrics were calculated from 20 different versions of five open-source Java software systems. The metrics were divided into three subsets: object-oriented metrics, centrality measures, and the combination of both. A total of 9 machine learning algorithms were used in their experiments; these were: Binary Logistic Regression (BLR), C5.0 Classification Tree, K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Partition

and Regression Tree, Random Forest (RF), Support Vector Machines (SVM), Linear Regression and Boosting with C5.0. In their results, they argue that algorithms based on decision trees (RF and C5.0) and techniques such as SVM have the best performance in classification using the combination of both metrics to determine classes prone to failure, obtaining an AUC value greater than 0.90, an ag-mean value greater than 85% and an F-measure greater than 75%.

On the other hand, Singh and Chhabra [15] propose a model for Cross-project software defect prediction using machine learning algorithms and structural code metrics. The proposed structural metrics were the following: (1) Extended Total Class Complexity (ETCC), which is used to measure class complexity, (2) Extended Coupling Among Classes (ECAC), which helps to evaluate the coupling between classes, (3) Extended Lack of Cohesion of Class Methods (ELCOCC), which determines the cohesion of class methods, (4) Extended Inherited Method Coupling Ratio (EIMCR), which quantifies the information inherited by the class, (5) Extended Overloaded Method Coupling Ratio (EOMCR), which helps measure method overload, and (6) Category of logic (COC). They also proposed the procedural metric of Weighted Path Complexity (WPC) code, to handle the shortcomings of the cyclomatic complexity. The metrics were calculated in 18 projects written in Java language. The machine learning algorithms used were: Decision Tree (DT), Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Adaptive boosting (AdaBoost), extreme gradient boost (XGBoost), and Artificial Neural Network (ANN). The categories for classification were assigned as defective or defects-free. The machine learning algorithm that achieved the highest average AUC in the study is XGBoost, with an average AUC of 0.76717.

The study by Canaparo et al. [7] aims to build defect prediction models by combining feature selection techniques and machine learning algorithms using various metrics from various projects. To do this, they used 2 datasets, the first the defect prediction dataset Eclipse with 5 software projects coded in Java that contains Chidamber and Kemerer metrics (CK) [22] such as Coupling between Objects (CBO), Depth of Inheritance Tree (DIT), Lack of Cohesion in Methods (LCOM), Number of Children (NOC), Response For Class (RFC), Weighted Methods for Class (WMC), 11 object-oriented metrics such as number of lines of code, number of methods, number of attributes, among others, and a binary tag. The second dataset corresponds to NASA's PROMISE repository related to 12 projects encoded in C and C++. The metrics they used were lines of code, metrics calculated from McCabe cyclomatic complexity, complexity metrics based on the number of operands and operators, among others. Different trait selection techniques were used, such as Principal Component Analysis (PCA), Backward Elimination, Forward Selection, and LASSO. Supervised Machine Learning algorithms such as Naïve Bayes(NB), Random Forest (RF), and unsupervised machine learning such as Neural Gas Clustering (GNG) and Spectral Clustering were implemented for defect prediction. The results showed that NB in combination with Backward Elimination achieved the highest performance in AUC with an average of 0.65, indicating it has a good ability to identify between

defective and defect-free classes, while Spectral Clustering achieved an average of 0.66 in AUC using all the features of the dataset and RF in combination with Backward an average of 0.61.

Khleel and Nehéz [23] present the training of different machine learning algorithms, such as Decision Tree (DT), Naïve Bayes (NB), Random Forest (RF) and Logistic Regression (LR). The algorithms were trained with 4 datasets from the PROMISE repository. The datasets used were JM1 and PC1, which contain metrics from codes written in C. Whereas, KC1, and KC2 contain code metrics in C++. These datasets contain metrics to measure the complexity of a program proposed by Halstead, such as: total number of operator and operands (N), Volume (V), Program length (L), Measure difficulty (D), Measure Intelligence (I), Measure Effort (E), Effort estimate (B), Time estimator (T); and McCabe complexity metrics, such as: the number of lines of code (LOC), cyclomatic complexity v(g), essential complexity ev(g), and design complexity iv(g). The results of their experiments showed that the DT and RF classifiers obtained the best results having 99% accuracy, 99% precision, 100% Recall for the JM1 and KC1 datasets, while KC2 showed an accuracy of 98 and 98% precision.

In Mohamed [24] the goal is to build a universal model based on the structure mainly of the Feedforward Neural Network (FFNN) to predict defects in a binary classification into defective and non-defective classes, using different datasets. These were collected from 12 projects where 38 versions of open-source object-oriented software coded in Java developed by the Apache Software Foundation were analyzed. The dataset analyzed Chidamber and Kemerer software complexity metrics, object-oriented metrics such as Weighted Methods for Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Object (CBO), Response for a Class (RFC), Lack of Cohesion in Methods (LCOM), and object-oriented metrics of cognitive complexity such as Method Complexity (MC), Coupling Weight for Class (CWC), Attribute Complexity (AC), Class Complexity (CLC), Code Complexity (CC) where the aim is to validate the association in the prediction of defects. Processing techniques were used, in the universal model 1 (UM1) the Min-Max normalization technique was applied, leaving the values between 0 and 1, the universal model 2 (UM2) the decile Rankin technique was used, which consists of mapping the values from 1 to 10 and taking these as thresholds in order to unify approximations, and a final universal model (UM0) where no processing was applied. In addition, several FFNN experiments were presented using different hidden layers and neurons, for the evaluation of the results they used 4 performance metrics where the values go from 0 to 1 and the highest indicate a better performance, UM1, UM2 and UM0 register the best maximum value of 0.69452 in Recall in the CC metrics, while CK metrics record a better result in all performance metrics.

Another study is presented by Deshpande et al. [1] where they proposed a model for evaluating the reliability of software through the early prediction of defects using Machine Learning algorithms. The metrics used were those of Chidamber and Kemerer (6): Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Objects (CBO), Response For a Class (RFC) and Lack of Cohesion in Methods (LCOM); Martins metrics

(2): Afferent Coupling (CA), Efferent Coupling (CE); QMOOD metrics (5): Number of public Methods (NPM), Data Access Metric (DAM), Measure of Aggregation (MOA), Measure of Functional Abstraction (MFA), Cohesion Among Methods (CAM); extended Chidamber and Kemerer metrics (4): Lack of Cohesion in method 3 (LCOM3), Inheritance coupling (IC), Total number of new/reddened methods to which all the inherited methods are coupled (CBM), Average Method per Class (AMC); size metrics (1): Line of code (LOC), Average McCabe cyclomatic complexity (MAX CC), Average McCabe cyclomatic complexity (AVG CC). The metrics were calculated on a total of 10 software projects, and all of these projects were developed in the Java programming language. The machine learning algorithms used were regression logistic (LR) and Decision Trees (DT). The algorithm that had the best results was LR, with an accuracy of 75%. In comparison, the DT showed an accuracy of 74.5%.

Table 1 summarizes the software metrics used to predict defects in each of the previous studies. The metrics were classified according to their purpose and object-oriented and traditional approach. In the object-oriented approach, metrics were identified in relation to complexity and size (WMC, CC, LOC, AMC, CIS) [1, 7, 21, 24]; inheritance (DIT, NOC, MFA, IC) [1, 7, 24]; cohesion (LCOM, LCOM4, CAM); coupling (CBO, CA, CE, CBM, RFC) [1, 7, 21, 24]; encapsulation and abstraction (DAM) [1], visibility (NPM) [1]; cognitive complexity metrics (MC, CWC, AC, CLC) [24]; as well as the proposal of new object-oriented structural metrics which seek to improve the limitations of the current ones (NOM, ETCC, ECAC, ELCOCC, EIMCR, EOMCR, COC, WPC) [15]. On the other hand, in the traditional approach, the complexity metrics of Halsted (N, V, L, D, I, E, B, T) and McCabe (EV G, MAX CC, AVG CC, V G, IV G) [1, 23] were used.

The most widely used metrics are WMC and CBO [1, 7, 21, 24], accompanied by LOC, DIT, NOC, IC, and RFC. In these studies, the use of metrics to generate models that help predict whether a class contains defects or even indicate the degree of severity of the defect has been proposed. However, as can be seen in Table 1, only [1] has proposed the combination of object-oriented metrics with other types of metrics. Therefore, our work, unlike previous studies, proposes the combination of various object-oriented metrics with software complexity metrics to predict defects, as well as the application of various algorithms to generate models that aim to detect software defects.

3 Methodology

The Inter-Industry Standard Process for Data Mining (CRISP-DM), developed by Wirth and Hipp [25], is recommended for experiment development due to its proven success in data mining, machine learning, and data science projects [26, 27]. Each stage of the process is detailed in the following subsections (Fig. 1).

Table 1 Comparison of studies that use software metrics to predict defects

Author		Ouellet et al. (2024)	Singh et al. (2024)	Canaparo (2022)	Khleel et al. (2021)	Mohamed (2020)	Deshpande et al. (2020)	Our work
Complexity and size metrics	WMC	x		x		x	x	x
	CC					x		
	LOC	x			x		x	x
	AMC						x	x
	ACC	x						
	CIS	x						
Inheritance metrics	DIT			x		x	x	x
	NOC			x		x	x	x
	MFA						x	x
	IC						x	x
Cohesion metrics	LCOM			x		x	x	x
	LCOM3						x	x
	CAM						x	x
Coupling metrics	CBO	x		x		x	x	x
	CA	x					x	x
	CE	x					x	x
	CBM						x	x
	RFC	x		x		x	x	x
Encapsulation and abstraction metrics	DAM						x	x
	MOA						x	x
Visibility metrics	NPM						x	x
Halsted complexity metrics	N				x			
	V				x			
	L				x			
	D				x			
	I				x			
	E				x			
	B				x			
	T				x			
McCabe complexity metrics	V(G)				x			
	EV(G)				x			
	IV(G)				x			
	MAX (CC)						x	x
	AVG(CC)						x	x

(continued)

Table 1 (continued)

Structural metrics	NOM	x						
	ETCC		x					
	ECAC		x					
	ELCOCC		x					
	EIMCR		x					
	EOMCR		x					
	COC		x					
	WPC		x					
Cognitive complexity metrics	MC					x		
	CWC					x		
	AC					x		
	CLC					x		

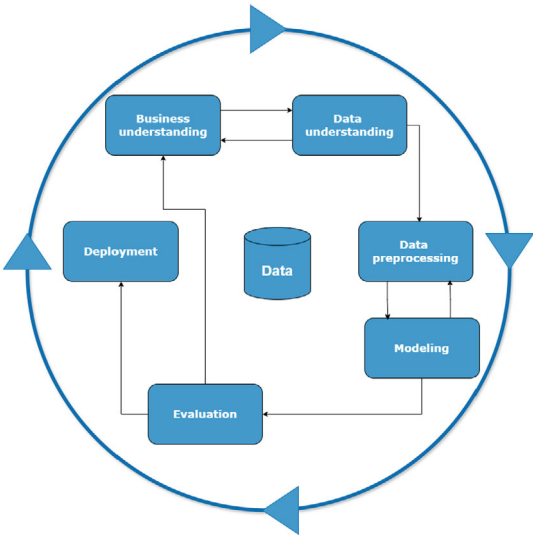


Fig. 1 CRISP-DM methodology phases [25]

3.1 Business Understanding

The goal of this research is to predict defects in software. To achieve the above, it is proposed to build a predictive model based on object-oriented metrics and software complexity metrics to classify software modules as defective or defect-free.

3.2 Data Understanding

The dataset used in this research is MJ12A. This is available for free online at http://purl.org/MarianJureczko/MetricsRepo. The MJ12A dataset is composed of three sets of metrics: (1) The metrics used to measure code understanding, maintenance, and reuse, proposed by Chidamber and Kemerer [22]; (2) The Quality Model for Object Oriented Design (QMOOD) metrics proposed by Bansiya and Davis [28], which are used to measure and evaluate aspects of software quality and user satisfaction; and finally 3) McCabe metrics, which are used to measure the complexity of a computer program [20].

The metric values come from 12 software projects written in Java. The calculations of the metrics were made using two computer programs: BugInfo, a tool that provides information on common defects and problems in software, helping developers and users to identify and resolve defects effectively. It is available online at https://www.si.edu/spotlight/buginfo and by Ckjm, a program that calculates Chidamber and Kemerer's object-oriented metrics by processing the bytecode of compiled Java files. This is open source and is available at: https://github.com/dspinellis/ckjm. The list and description of the metrics is presented in Tables 2, 3, 4, 5, 6 and 7. These were collected from the publication by Jureczko and Spinellis (2010) on the MJ12A dataset and were classified according to their type into: complexity and size metrics, inheritance metrics, cohesion metrics, coupling metrics, encapsulation and abstraction metrics, visibility metrics.

The dataset contains 98,148 records with 29 attributes. Of these 29 attributes, 20 correspond to complexity and object-oriented software metrics, which belong to various categories. These are presented and described in Tables 2, 3, 4, 5, 6 and 7. The rest of the attributes are bugs, which specify the number of bugs identified in the software; it has been transformed into a class-type attribute. If the number of bugs is equal to zero, it is assigned the "defect-free" class, while if the number of bugs is greater than or equal to 1, it is assigned the "defective" class. Other attributes contained in the dataset are project name, version, and Java classes, which were not considered to perform the experiments. The dataset used in our experiment consists of 20 software metrics and the class (defective and defect-free). The dataset is distributed as follows: 79,849 records corresponding to the "defect-free" class (81.35%) and 14,299 to the "defective" class (18.64%).

3.3 Data Preprocessing

3.3.1 Normalization

With the aim of homogenizing the data and avoiding biases to machine learning algorithms. In this work, Z-Score normalization was used, which consists of transforming the units of measurement into new data which have a mean equal to 0 and a standard deviation equal to 1. The equation used to normalize the data was

Table 2 Complexity and size metrics

Name of metrics	Metric description
Class-weighted methods (WMC)	The value of WMC is equal to the number of methods in the class (assuming unitary weights for all methods)
Lines of code (LOC)	Based on Java binary code. It is the sum of the number of fields, methods, and statements in each method of the class
Average method complexity (AMC)	Calculates the average size of methods in a class, measured in number of java binary codes in the method
McCabe cyclomatic complexity (CC)	Number of different paths in a plus-one method
Avg (CC)	Arithmetic mean of CC in the methods of the class
Max (CC)	Highest CC value among the methods in the class

Table 3 Inheritance metrics

Name of metrics	Metric description
Depth of inheritance tree (DIT)	Levels from root to class
Number of children (NOC)	Measures the number of immediate descendants of the class
Measure of functional abstraction (MFA)	It is the ratio between the number of methods inherited by a class and the total number of accessible methods
Inheritance coupling (IC)	Measures the number of parent classes to which a class is coupled. A class is coupled to its parent and one of its inherited methods is functionally dependent on new or redefined methods to which all inherited methods are coupled

Table 4 Cohesion metrics

Name of metrics	Metric description
Lack of cohesion in methods (LCOM)	Counts the method pairs in a class that do not share access to at least one common field. This is calculated by subtracting the number of pairs that share access from those that do not
Lack of cohesion in methods (LCOM3)	Variant that calculates cohesion in methods
Class method cohesion (CAM)	Calculates the relationship between methods of a class based on the parameter list

Table 5 Coupling metrics

Name of metrics	Metric description
Coupling between classes of objects (CBO)	Represents the number of classes coupled to a specific class (includes both efferent and afferent couplings). These couplings occur through method calls, field access, inheritance, method arguments, return types, and exceptions
Afferent coupling (Ca)	Measures the number of classes that depend on the measured class
Efferent coupling (Ce)	Measures the number of classes on which the measured class depends
Coupling between methods (CBM)	Coupling between methods
Response for a class (RFC)	Measures the number of different methods that can be executed when an object of that class receives a message. It includes internal methods and those called within the body of the class's methods

Table 6 Encapsulation and abstraction metrics

Name of metrics	Metric description
Data access metrics (DAM)	It is the ratio between the number of private and protected attributes and the total number of attributes declared in the class
Measure of aggregation (MOA)	Measures the degree of part-whole relationship in a class based on the number of class fields that are of user-defined types

Table 7 Visibility metrics

Name metrics	Metric description
Number of public methods (NPM)	Counts all methods of a class that are declared as public

$$Z\text{-Score} = \frac{x - \mu}{\sigma} \tag{1}$$

where Z-Score is the standardized score, x is the data to be analyzed, μ is the average of the distribution of the data, and σ is the dispersion of the data with respect to the mean.

To leave the standardized score on a numerical scale from 0 to 1, the percentile rank operation is used, given by the following Eq. 2.

$$PRank = \left(\text{trunk}\left(\frac{\text{rank}(x)}{\text{length}(x)}\right)\right) \times 100 \tag{2}$$

3.3.2 Data Balancing

Whereas the MJ12A dataset is not uniformly distributed, as it is distributed as follows: 14,299 records belonging to the "defective" class and 79,849 to the "defect-free" class. It was necessary to perform a data balance, that is, to balance the "defective" class, penalizing the "defect-free" class during training. The balance of classes was carried out with the SMOTE (Synthetic Minority Over-Sampling Technique) function in R language. This technique creates new synthetic examples by interpolating between the examples of the "defective" class and their nearest neighbors, which helps to improve the generalization of the classifier [29].

SMOTE was chosen over other balancing techniques because, unlike methods such as oversampling and downsampling, it generates synthetic data instead of merely duplicating minority class observations or removing majority class observations [30].

The configuration used by SMOTE in our experiment was as follows. The number of neighbors used for the generation of new instances was K = 5. The Euclidean distance metric was used to calculate the nearest neighbors. The dataset was balanced by increasing the minority class by 450% therefore the dup_size parameter was 4.5.

3.4 Modeling

Several machine learning algorithms were used to generate predictive models for the estimation of software defects using object-oriented and software complexity metrics. Each of the algorithms used is described below.

3.4.1 K-Nearest Neighbors (K-NN)

The algorithm consists of having N cases, each one is characterized by n predictor variables $X_1, \ldots, X_n$ and a variable to predict class C. The new case to be classified is denoted by $x(X_1, \ldots, X_n)$. The process carried out in the algorithm is as follows, the values of the attributes are used to calculate the distance of the new case to be classified with respect to the N cases already characterized. Various measures of similarity can be used to calculate distances, such as Euclidean distance, Manhattan, to name a few. Once the distances are calculated, they are ordered from smallest to highest. Subsequently, a value of K is defined to select the K closest neighbors to the new element to be classified. There are several heuristics that have been proposed to calculate the value of K, such as: (1) use coprime class and k combinations, (2) choose a K that is greater or equal to the number of classes plus one, and (3) choose a K that is low enough to avoid noise [31]. In this work, we calculate various K values using heuristic number one and Euclidean distance was used, as there are studies that show that this distance has shown good results in certain contexts [32, 33].

The values for K were: 3, 5, 7, 9, 11 and 15. From the nearest neighbors K they get their classes. The most frequent class C among the neighbors is the one assigned to this new element [34].

3.4.2 Naïve Bayes (NB)

This algorithm is based on Bayes' theorem. Which provides a way to calculate the subsequent probability using the equation (Ec).

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)} \tag{3}$$

Naïve Bayes' algorithm assumes that the effect of the value of a predictor (x) in a class (c) is independent of the values of other predictors [35]. This assumption is called conditional class independence. The subsequent probability is calculated with the following equation

$$P(c|x) = P(x_1|c) \cdot P(x_2|c) \cdot \ldots \cdot P(x_n|c) \cdot P(c) \tag{4}$$

where: $P(c|x)$ is the posterior probability of class (target) given predictor (attribute), $P(c)$ is the prior probability of class, $P(x|c)$ is the likelihood which is the probability of the predictor given class and $P(x)$ is the prior probability of the predictor.

3.4.3 Random Forest (RF)

The random forest algorithm is based on decision trees and the bagging technique. Bagging consists of randomly selecting individuals from a dataset using sampling with replacement to create different datasets called bootstrap samples. From each of these bootstrap samples, a decision tree is generated. Each tree depends on independently tested random values with the same distribution for each of these. Finally, the new data are predicted using majority voting. The hyperparameters used in the experiments were 100, 300 and 500 trees(ntree), with a depth of 3 (mtry).

3.4.4 Support Vector Machine (SVM)

The vector support machine (SVM) algorithm first maps the entry points to a feature space of a larger dimension and finds a hyperplane that separates them and maximizes the margin between classes in this space [36]. Without any knowledge of mapping, the SVM finds the optimal hyperplane using the point product with feature space functions called kernels. The solution of the optimal hyperplane can be written as the combination of a few entry points that are called support vectors. Kernel functions are mathematical functions that are used in SVM, examples of these functions are

linear, polynomial, tangential and radial. These functions are what make it possible to convert what would be a nonlinear classification problem in the original dimensional space to a simple linear classification problem in a larger dimensional space [36]. The hyperparameters used in the experiments were a radial kernel, with a cost equal to 1.

3.4.5 Multilayer Perceptron (MLP)

The perceptron is an artificial neural network model that is characterized by being composed of multiple units called simple perceptrons. Its architecture is organized into several hierarchical layers: an input layer, one or more hidden layers, and an output layer. In the input layer, data is received in the form of a vector $x \in \mathbb{R}^d$, where d represents the number of features in the dataset. Each neuron in a layer performs two key operations: weighting and addition ($z = w^T\ x + b$) and applying an activation function ($a = f(z)$). Where w is the weights associated with the connections, b is a bias, and f is a nonlinear function such as the sigmoid, ReLU (Rectified Linear Unit) or tanh, which introduces nonlinearity into the model [37]. In our experiments a Multiplayer Perceptron (MLP) was generated with 2 dense layers, 128 neurons in the first layer, 64 neurons in the second, both layers use RELU and the output layer was softmax as an activation function, dropout rate of 0.3 for both layers, a batch of 32, learning rate of 0.001, momentum of 0.2, and Adam was used as an optimizer.

3.5 Model Evaluation and Validation

There are a number of metrics that help us evaluate the performance of machine learning algorithms. The ones used in our experiments are presented and described in Table 8, where TP is true positives, FP is false positives, TN is true negatives, FN is false negatives, Pc is the accuracy of the real classifier, Pr is the accuracy of the random classifier. Additionally, for some metrics such as the statistical kappa, the interpretation scale presented in Table 9 is used.

To perform the evaluation and validation of the models, the dataset balanced with the SMOTE technique was partitioned into 80% for training and 20% for testing. The 80% of the training was applied the 10 cross-validation technique, which consists of dividing the dataset into 10 equal parts. Nine parts are used to train the model and the tenth for validation, alternating the latter in each iteration until all parts are covered [38].

Table 8 Metrics for evaluating the performance of machine learning algorithms

Metric	Acronym	Definition	Equation
Precision	Pr	The accuracy of an algorithm refers to how accurately it performs its task compared to the correct or expected result	$Pr = \frac{TP}{TP+FP}$
Accuracy	Acc	The accuracy of an algorithm refers to its ability to produce correct and precise results in relation to the objective it seeks to achieve	$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$
F1-Score	F1	This metric is used to evaluate the precision and recall of an algorithm using a single value. It is calculated as the harmonic mean of the precision and recall	$F1 = 2 \times \left(\frac{Pr \times R}{Pr+R}\right)$
Recall	R	The recall of an algorithm refers to the ability of the algorithm to correctly identify all relevant cases within a data set	$R = \frac{TP}{TP+FN}$
Sensitivity	SE	It is a metric that measures the algorithm's ability to correctly identify positive cases	$SE = \frac{TP}{TP+FN}$
Specificity	SP	It is a metric that measures the algorithm's ability to correctly identify negative cases	$SP = \frac{TN}{TN+FP}$
Geometric mean	G-Mean	It evaluates the balance between the True Positive Rate (TPR) and the True Negative Rate (TNR), ensuring that both are maximized in a balanced manner	$\text{G-Mean} = \sqrt{\text{TPR} \cdot \text{TNR}}$
Kappa statistic	k	It is a metric used to compare the performance of a classifier with that of a classifier that makes predictions based solely on chance	$k = \frac{Pc-Pr}{1-Pr}$

Table 9 Kappa scale for interpretation [38]

k value	Description
<0	Less than chance performance
0.01–0.20	Slightly good
0.21–0.40	Fair performance
0.41–0.60	Moderate performance
0.61–0.80	Substantially good performance
0.81–1.00	Near perfect performance

3.6 Experimental Setup

Three experiments were conducted for predicting software defects: (1) the algorithms were trained with software complexity metrics proposed by McCabe [20] and Chidamber and Kemerer [22]. The resulting set of metrics was: WMC, DIT, NOC, CBO, RFC, LCOM, MAX.CC, AVG.CC; (2) the algorithms were trained with the Quality Model for Object-Oriented Design (QMOOD) metrics proposed by Bansiya and Davis [28]. These metrics were: CA, CE, NPM, LOC, DAM, MOA, MFA, CAM, IC, CBM, AMC; and finally, (3) the algorithms were trained with a combination of software complexity and object-oriented metrics, having a total of 20 metrics. The final list was: WMC, DIT, NOC, CBO, RFC, LCOM, MAX.CC, AVG.CC, CA, CE, NPM, LOC, DAM, MOA, MFA, CAM, IC, CBM, AMC.

In all experiments, the following machine learning algorithms were trained: K-nearest neighbors (KNN), Naïve Bayes (NB), Random Forest (RF), Vector Support Machines (SVM), and a Multilayer Perceptron (MLP).

To implement the machine learning algorithms, an Asus brand computer, model ASUS TUF Gaming $F15\ FX506LH\_FX506LH$, with Intel(R) Core(TM) $i5-10300H\ CPU@2.50GHz$ 2.50 GHz, 32.0 GB RAM. The algorithms were implemented in RStudio 2024.04.2 with version 4.4.1 of the R language. In addition, the packages listed in Table 10 were used.

Table 10 Libraries used to implement machine learning algorithms

Library	Version	Description
Class	7.3–22	Package that includes classification functions. This was used to implement KNN
Caret	6.0–94	It was used to create training and test sets for model validation
e1071	1.7–16	Package that provides machine learning and statistics functions. It was used to implement Naïve Bayes and Support Vector Machines (SVM)
Neuralnet	1.44.2	It is used for the implementation of neural networks
Randomforest	4.7–1.2	Library that implements the Random Forest algorithm used for classification
pROC	1.18.5	Package used to visualize Roc curves, and helps us to evaluate and compare the predictive capability of the models
Keras	2.15.0	Package for the implementation of neural networks
Smotefamily	1.14.0	Package for data generation

4 Results

This section presents the best performance results of each algorithm used in the experiments conducted and described in Sect. 3.6.

4.1 Training Algorithms with Complexity Metrics

Table 11 presents the performance of the algorithms when they are trained with software complexity metrics. It shows that the KNN algorithm obtained the best performance with an accuracy of 0.8362, a precision of 0.8026, F1-Score of 0.8332, recall of 0.8663, and G-Mean of 0.7994. These results were obtained using a K equal to 3 and an Euclidean distance. Further, the algorithm with the lowest performance was Naïve Bayes with an accuracy of 0.6260, and a sensitivity of 0.5016 for identifying the defective class.

4.2 Training Algorithms with Software Object Oriented Metrics

The performance of machine learning algorithms trained on object-oriented metrics is presented in Table 12. The algorithm with the best performance is KNN with K equal to 3 and Euclidean distance, having a performance with an accuracy of 0.8076, a sensitivity of 0.8557, an F1-Score of 0.8309, and G-Mean of 0.8013. The algorithm with the lowest performance was Naïve Bayes with an accuracy of 0.6096, a sensitivity of 0.5505, and G-Mean of 0.6138.

Table 11 Algorithm performance using complexity metrics

ML algorithm	Acc.	Pr.	R	F1	SE	SP	k	G-Mean
KNN	**0.8362**	**0.8026**	**0.8663**	**0.8332**	**0.7377**	**0.8842**	**0.6162**	**0.7994**
NB	0.6228	0.6246	0.5216	0.5684	0.7193	0.5216	0.2430	0.6125
RF	0.8095	0.8164	0.7698	0.7924	0.8451	0.7698	0.6167	0.8065
SVM	0.6750	0.6760	0.6722	0.6722	0.6750	0.3330	0.3441	0.6835
MLP	0.6987	0.6990	0.6990	0.6970	0.6990	0.3080	0.3924	0.6990

Table 12 Performance of algorithms using object-oriented metrics

ML algorithm	Acc.	Pr.	R	F1	SE	SP	k	G-Mean
KNN	**0.8355**	**0.8076**	**0.8557**	**0.8309**	**0.7505**	**0.8700**	**0.6710**	**0.8013**
NB	0.6211	0.6096	0.5505	0.5785	0.6844	0.5505	0.2361	0.6138
RF	0.7560	0.7530	0.7190	0.7360	0.7190	0.2110	0.594	0.7190
SVM	0.7267	0.7260	0.7270	0.7260	0.7270	0.2770	0.4506	0.7270
MLP	0.7270	0.7280	0.7270	0.7260	0.7270	0.2800	0.4495	0.7270

4.3 Training Algorithms with Complexity Metrics and Object Oriented Software Metrics

Table 13 shows the performance results of machine learning algorithms trained with complexity and object-oriented metrics. The algorithm with the best performance is Random Forest (RF) with 100 trees (ntree) and 3 randomly selected features considered when splitting a node (mtry). The accuracy of RF was 0.8723, a precision of 0.8552, a Recall of 0.8785, an F1-Score of 0.8667, and G-Mean of 0.8726. The algorithm with the lowest performance was Naïve Bayes with a precision of 0.6200, a recall of 0.5640, and G-Mean of 0.6240.

5 Discussion

In our experiments, as in the previous studies presented in the related work by Deshpande et al. [1], Ouellet and Badri [21], Khleel and Nehéz [23] provide evidence that software metrics are a key element for the detection of software defects. Therefore, its use in the training of machine learning algorithms for the creation of predictive models is very useful. The results obtained in our experiments show that the combination of object-oriented metrics and software complexity allows more accurate identification of defects in object-oriented software, not using object-oriented metrics alone. In this regard, Singh and Chabbra [15] argue that object-oriented metrics have

Table 13 Performance through object-oriented and software complexity metrics

ML algorithm	Acc.	Pr.	R	F1	SE	SP	k	G-Mean
KNN	0.8371	0.8009	0.8718	0.8349	0.7412	0.8875	0.6162	0.8038
NB	0.6307	0.6200	0.5640	0.5906	0.6905	0.5640	0.2557	0.6240
RF	**0.8723**	**0.8552**	**0.8785**	**0.8667**	**0.8668**	**0.8786**	**0.7443**	**0.8726**
SVM	0.7528	0.7530	0.7530	0.7520	0.7530	0.252	0.5225	0.7530
MLP	0.7275	0.7280	0.7280	0.7270	0.7280	0.2770	0.4715	0.7280

limitations in capturing in-depth information about the complexity of classes, as do structural complexity metrics such as McCabe's. However, the combination of both metrics increases defect prediction accuracy by 5.26% compared to using software complexity metrics alone and 4.76% when only object-oriented metrics are used. Even our results offer better performance than previous work that combines complexity and object-oriented metrics using software developed in Java, for example, in Deshpande et al. [1], who obtained a performance of 75% using logistic regression (RL). In our results, the Random Forest (RF) algorithm was the one that presents the best performance for defect prediction training with object-oriented and complexity metrics. It obtained an accuracy of 0.8723, an precision of 0.8552, a recall of 0.8785, F1-score of 0.8667, a specificity of 0.8668, Kappa Statistic of 0.7443 and G-Mean of 0.8726. The results obtained with RF indicate that the generated model is capable of detecting the defective and defect-free classes. The value of the kappa statistic (k) indicating that RF performance is not only high but also consistent beyond chance.

On the other hand, NB (Naïve Bayes) consistently shows the lowest values in Accuracy, Precision, Kappa Statistic, G-Mean, and F1-Score. The discrepancy between accuracy and Kappa Statistic (k) in NB highlights that its performance may be more erratic or biased. This may be because software metrics cannot be used as conditionally independent predictor variables; just as the NB algorithm assumes. Because the metrics together provide a complete view of the quality of design, implementation, and maintainability of an object-oriented system.

The developed models can be adapted for languages such as C++, C#, Python, which follow the principles of object-oriented programming. Since, complexity metrics, such as cyclomatic complexity, nesting depth, and coupling, are universal principles that apply to any programming language. These metrics can be used to evaluate code quality and predict defects in various languages, such as Python, C++, C# among others. As well as some object-oriented metrics, such as cohesion, inheritance, and encapsulation.

Additionally, the machine learning algorithms used in our experiments, such as KNN, SVM, RF y MLP, are independent of programming language. The algorithms can be applied to data from different programming languages and development environments, as long as the appropriate metrics are available.

5.1 Limitations

The generation of the predictive models presented in this chapter were explorer using only the Java programming language. It would be interesting to investigate their performance with other object-oriented languages.

6 Conclusions and Future Work

This chapter presents a comparative analysis of various traditional machine learning algorithms trained with software metrics. Three different experiments were carried out, training the algorithms with software complexity metrics, with object-oriented metrics and their combination. This with the aim of determining which combination of machine learning algorithm and set of metrics allows predicting software defects with greater accuracy. The results indicate that combining object-oriented and complexity metrics yields better results than using a single type of metrics. As a future work, with the aim of generating more robust models, it is proposed to investigate whether the metrics used for defect prediction in Java are applicable and effective for software developed in other object-oriented languages, such as Python, C++, C#, among others. Expand the set of metrics, and perform a trait selection process with the goal of creating generalizable models. As well as research on the limitations of software metrics to determine software complexity.

Declarations
No conflict of interest.

Acknowledgements To Consejo Nacional de las Humanidades, las Ciencias y las Tecnologías (CONAHCyT) for the scholarship awarded to the first author for the completion of his master's degree studies in science (CVU:1165525).

References

1. Deshpande, M.B.S., Kumar, B., Kumar, A.: Object oriented design metrics for software defect prediction: An empirical study. TEST Eng. Manage **83**(1), 10501–10518 (2020)
2. Cass, S.: The top programming languages: Our latest rankings put python on top-again-[careers]. IEEE Spectrum **57**(8), 22–22 (2020)
3. Durelli, V.H., Durelli, R.S., Borges, S.S., Endo, A.T., Eler, M.M., Dias, D.R., Guimarães, M.P.: Machine learning applied to software testing: A systematic mapping study. IEEE Transactions on Reliability **68**(3), 1189–1212 (2019)
4. Singh, S.K., Singh, A.: Software Testing, 1st edn. Vandana Publications, Delhi, India (2012)
5. Software Engineering Terminology, I.S.G.: IEEE Std 610.12–1990 (1990)
6. Dustin, E.: Effective Software Testing: 50 Specific Ways to Improve Your Testing, 1st edn. Pearson Education India, Delhi, India (2003)
7. Canaparo, M., Ronchieri, E., Bertaccini, G.: Software defect prediction: A study on software metrics using statistical and machine learning methods. In: Proceedings of Science, vol. 415, pp. 21–25 (2022)
8. MicroFocus: MicroFocus. Obtenido de Unified Functional Testing (UFT) (2024). https://www.microfocus.com/es-es/products/unified-functional-automated-testing/overview
9. SeleniumHQ: SeleniumHQ. Obtenido de: https://www.seleniumhq.org (2024). https://www.seleniumhq.org
10. EggPlant: EggPlant. Obtenido de: https://eggplant.io/ (2024). https://eggplant.io/
11. Ranorex: Ranorex. Obtenido de: https://www.ranorex.com/ (2024). https://www.ranorex.com/

12. SmartBear: SmartBear Software. Obtenido de TestComplete (2024). https://smartbear.com/product/testcomplete/overview/
13. Microsoft: Visual Studio Test Professional. Obtenido de: https://visualstudio.microsoft.com/es/vs/test-professional/ (2024). https://visualstudio.microsoft.com/es/vs/test-professional/
14. Saeed, M.S., Saleem, M.: Cross project software defect prediction using machine learning: a review. International Journal of Computational and Innovative Sciences **2**(3), 35–52 (2023)
15. Singh, M., Chhabra, J.K.: Machine learning based improved cross-project software defect prediction using new structural features in object oriented software. Applied Soft Computing **165**, 112082 (2024)
16. Bishop, C.: Pattern Recognition and Machine Learning, vol. 16, pp. 140–155 (2006). https://doi.org/10.1117/1.2819119
17. Shafiq, S., Mashkoor, A., Mayr-Dorn, C., Egyed, A.: A literature review of using machine learning in software development life cycle stages. IEEE Access **9**, 140896–140920 (2021)
18. Salam, M.A., Abdel-Fattah, M., Moemen, A.A.: A survey on software testing automation using machine learning techniques. Int. J. Comput. Appl **183**(51), 12–19 (2022)
19. Kanmani, S., Uthariaraj, V.R., Sankaranarayanan, V., Thambidurai, P.: Object-oriented software fault prediction using neural networks. Information and software technology **49**(5), 483–492 (2007)
20. McCabe, T.J.: A complexity measure. IEEE Transactions on software Engineering (4), 308–320 (1976)
21. Ouellet, A., Badri, M.: Combining object-oriented metrics and centrality measures to predict faults in object-oriented software: An empirical validation. Journal of Software: Evolution and Process **36**(4), 2548 (2024)
22. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Transactions on software engineering **20**(6), 476–493 (1994)
23. Khleel, N.A.A., Nehéz, K.: Comprehensive study on machine learning techniques for software bug prediction. International Journal of Advanced Computer Science and Applications **12**(8) (2021)
24. Mohamed, F.A., Salama, C.R., Yousef, A.H., Salem, A.M.: A universal model for defective classes prediction using different object-oriented metrics suites. In: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 65–70 (2020). IEEE
25. Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, vol. 1, pp. 29–39 (2000). Manchester
26. Clark, A.: The machine learning audit-crisp-dm framework. Isaca Journal **1**, 42–47 (2018)
27. Schröer, C., Kruse, F., Gómez, J.M.: A systematic literature review on applying crisp-dm process model. Procedia Computer Science **181**, 526–534 (2021)
28. Bansiya, J., Davis, C.G.: A hierarchical model for object-oriented design quality assessment. IEEE Transactions on software engineering **28**(1), 4–17 (2002)
29. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority oversampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
30. Taban, R., Nunes, C., Oliveira, M.R.: Rm-smote: A new robust balancing technique. Expert Systems with Applications **210**, 118045 (2023)
31. Kirk, M.: Thoughtful Machine Learning with Python: A Test-Driven Approach. O'Reilly Media, Inc., Sebastopol, CA (2017)
32. Lopes, N., Ribeiro, B.: On the impact of distance metrics in instance-based learning algorithms. In: Pattern Recognition and Image Analysis: 7th Iberian Conference, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings 7, pp. 48–56 (2015). Springer
33. Todeschini, R., Ballabio, D., Consonni, V., Grisoni, F.: A new concept of higher-order similarity and the role of distance/similarity measures in local classification methods. Chemometrics and Intelligent Laboratory Systems **157**, 50–57 (2016)
34. Moujahid, A., Inza, I., Larranaga, P.: Tema 5. clasificadores k-nn. Departamento de Ciencias de la Computación e inteligencia artificial, Universidad del País Vasco-Euskal Herriko Unibertsitatea **3**, 1 (2019)

35. Sayad, S.: An introduction to data science. Toronto: University of Toronto (2010)
36. Pisner, D.A., Schnyer, D.M.: Support vector machine. In: Machine Learning, 1st edn., pp. 101–121. Elsevier, Amsterdam, Netherlands (2020). Chap. 5
37. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning cambridge. MA: MIT Press http://www.deeplearningbook.org (2016)
38. Zacharski, R.: A Programmer's Guide to Data Mining| The Ancient Art of the Numerati', Guidetodatamining. com, 2015

Application of IMU Sensors and AI for Infrastructure Monitoring

M. Fernanda López-Barajas, Julio C. Rodríguez-Quiñonez, Wendy Flores-Fuentes, Jonathan J. Sanchez-Castro, Jorge Alejandro Valdez-Rodríguez, Moises J. Castro-Toscano, and Oleg Sergiyenko

Abstract IMU (Inertial Measurement Unit) sensors were initially used in space and military contexts. However, with the advancement of technology, new generations of these sensors have emerged and are now finding applications in several fields. The development of Artificial Intelligence (AI) has been an essential part of this progress, driving improvements in the ability of IMUs to adapt to complex and variable environments, mainly thanks to the integration of neural networks. This synergy between IMU sensors and AI has allowed their use in applications such as infrastructure monitoring, covering the structural health of buildings, street mapping, and detecting irregularities in road infrastructure. In these areas, methods such as the use of Support Vector Machines (SVM) or classification and regression in Structural Health Monitoring (SHM) systems and real-time structural damage detectors using one-dimensional Convolution Neural Networks (CNN), and Artificial Neural Networks (ANN), among other applications, stand out. Finally, a case study is presented in which a CNN is used to classify speed bumps from acceleration data obtained

M. F. López-Barajas · J. C. Rodríguez-Quiñonez (✉) · W. Flores-Fuentes · J. J. Sanchez-Castro · M. J. Castro-Toscano
Facultad de Ingeniería, Universidad Autonoma de Baja California, Mexicali, BC, Mexico
e-mail: julio.rodriguez81@uabc.edu.mx

M. F. López-Barajas
e-mail: maria.lopez.barajas@uabc.edu.mx

W. Flores-Fuentes
e-mail: flores.wendy@uabc.edu.mx

J. J. Sanchez-Castro
e-mail: jonathan.sanchez11@uabc.edu.mx

M. J. Castro-Toscano
e-mail: moises.castro@uabc.edu.mx

J. A. Valdez-Rodríguez
School of Engineering, CETYS Universidad, Mexicali, BC, Mexico
e-mail: jorge.valdez@cetys.mx

O. Sergiyenko
Instituto de Ingeniería, Universidad Autonoma de Baja California, Mexicali, BC, Mexico
e-mail: srgnk@uabc.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_24

by an IMU sensor. The CNN inputs consist of windows generated from the inertial sensor readings. The system proved to be efficient for real-time speed bump classification, achieving 91% accuracy in training tests. During real-time experimentation, the network achieves 90% accuracy when applying a probability threshold of 0.9 for the "speed bump" class.

1 Introduction

Inertial Measurement Unit (IMU) sensors are electronic devices that integrate accelerometers, gyroscopes, and magnetometers [1]. These components measure acceleration and angular velocity and detect magnetic heading. By applying appropriate signal processing, the position and orientation of an object in space can be estimated from its measurement. This estimation is based on the values of velocity and angular rotation.

IMU sensors have many advantages, including low cost, lightweight, fast response, and autonomous operation. However, their main advantage is that each component provides information independently; combined, they create a robust and efficient system for navigation, platform stabilization, motion tracking, autonomous vehicle control, and structure health monitoring applications.

The raw data obtained from an IMU sensor are illustrated in the diagram in Fig. 1. This includes measurements of acceleration, angular velocity, and magnetic field. After calibration, the orientation and angular velocity in the pitch, roll, and yaw axes can be determined to estimate the object's position in three-dimensional space.

While IMU sensors offer appealing advantages, they also come with significant challenges. Although they provide high accuracy in the short term, long-term position and orientation estimation can be problematic. Over time, position and orientation calculations are susceptible to accumulating errors, which can result in drift and increased sensitivity to noise [2]. In addition, certain IMUs can be affected by interference from external magnetic fields. However, the Earth's magnetic field provides

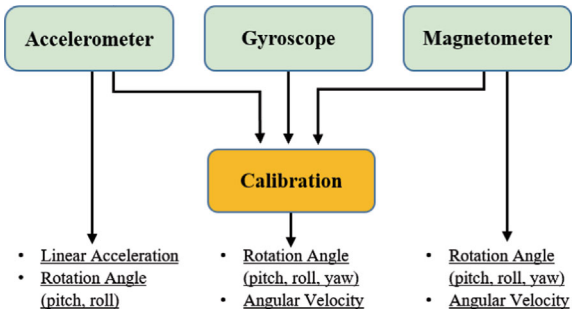


Fig. 1 Obtained measurements from a 9-DOF IMU sensor, integrating an accelerometer, gyroscope, and magnetometer

a standard, accurate, and reliable reference that magnetometers or compasses can use to improve orientation determination [3], partly mitigating other IMU components' challenges.

IMUs, depending on their degrees of freedom (DOF), can leverage these additional references to improve accuracy. For example, an IMU containing a three-axis accelerometer (x, y, z) and a three-axis gyroscope (yaw_g, pitch_g, roll_g) provides a system with six degrees of freedom [4]. Although an IMU with these two sensors is robust to external magnetic fields, estimating position based solely on accelerometer and gyroscope data can introduce cumulative drift, affecting long-term accuracy. Using a magnetometer (yaw_m, pitch_m, roll_m) enhances the precision of orientation estimation in IMUs. These three degrees of freedom for each sensor result in a system with nine degrees of freedom [4]. While the magnetometer assists in calibrating the gyroscope and improving accuracy, its performance may be impacted by external disturbances when used near ferromagnetic materials. We can observe in Fig. 2 an example of an IMU sensor with its coordinate system used (x, y, z) for the accelerometer and the rotation angles (ψ, θ, ϕ)corresponding to the gyroscope and the magnetometer, where ψ represents the yaw, θ the pitch, and ϕ the roll.

Due to their size, cost, and high energy consumption, IMUs' early applications were limited. They were primarily restricted to large systems, with an initial focus on the military industry, specifically missile systems. The evolution of IMU sensors can be divided into three main phases. The first phase spans from 1940 to 1960 when mechanical gyroscopes were used. Given their initial success, inertial sensors have continued to advance and improve. One of the most significant improvements was size; early inertial devices weighed around 3 kg, limiting their use to large applications [5]. For this reason, inertial devices were initially used in the guidance system. The second phase covers the period from 1960 to 1980, during which platform gyroscopes were integrated. Although electromechanical devices began to be developed in the 1960s, microelectromechanical systems (MEMS) were introduced in the 1980s [6], marking the beginning of the era of micromachining and nanotechnology. Finally, from 1980 to the present, microelectromechanical (MEMS) gyroscopes have been developed [1]. The development of this technology has been widely recognized by countries with advanced industrial sectors, considering it a key driver of economic

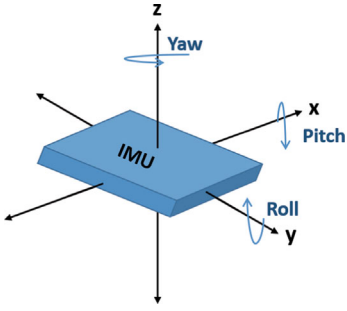


Fig. 2 Coordinate system and the rotation angles in IMU sensors

growth. Over the years, their weight and size have decreased, and today, an IMU sensor can weigh less than 5 g, thanks to advancements in MEMS technology.

Integrating accelerometers, gyroscopes, and magnetometers into a single compact device has been crucial for the exponential increase in demand for MEMS-based IMUs. This trend is driven by the need to develop smaller electronic systems and devices, such as cell phones, laptops, and autonomous systems.

Thanks to advances in size, weight, compatibility, and cost, modern commercial IMU sensors are now used across a wide range of fields, including robotics, autonomous navigation, electronics, military, aerospace, agriculture, telecommunications, computing, automobiles, transportation, sport, medicine, and Structural Health Monitoring (SHM). Specific examples include personal electronic devices [7], uncrewed aerial and ground vehicles [8], performance assessment sports [9], airbag systems [10], depth measurement in well drilling [11], DNA analysis [12], missile systems [13], and monitoring and analysis of structural integrity in construction [14–17].

IMU sensors have evolved significantly and are integrating into new applications. One area where they have made a notable impact is in Structural Health Monitoring (SHM). Thanks to their high precision, these sensors are helpful for early fault detection and optimizing structural maintenance. The technological development of SHM emerged with the primary goal of reducing maintenance costs in aircraft structures by implementing autonomous damage detection systems that continuously monitor the structure's integrity. This advancement took place during the 1990s [18]. Now, the SHM has begun to be implemented for health monitoring in various structures beyond the aerospace industry.

With the continued advancement of IMU technology and the increasing incorporation of artificial intelligence, these combined tools are expected to expand their impact and effectiveness in various fields. Although integrating AI with IMUs is still an emerging field, the combined results show great potential. In this chapter, different methods are discussed, such as the use of Artificial Neural Networks (ANN) [17, 19, 20], 1-dimensional Convolution Neural Networks (CNN) [21], and Least-squares support-vector machines (LS-SVM) [22]. Finally, Sect. 4 presents a case study focused on the real-time detection of speed bumps. It uses data windows with acceleration measurements from an IMU sensor as inputs in a CNN to classify the presence of a speed bump. The procedure and the results obtained will be explained.

2 Impact on Infrastructure Monitoring Using IMUs

Infrastructure monitoring is an ongoing field with multiple applications and relevance in the construction industry, considerably impacting Structural Health Monitoring (SHM). Conventional methods for damage assessment involve direct and periodic visual inspection of structures, which is a difficult task for civil engineers because they have to detect, locate, and assess possible structural damage through visual cues on site. And so, incorporating sensing technology has been a desired implementation

for SHM. In the literature, a commonly used sensing device for SHM is the Inertial Measuring Unit (IMU) and, specifically, the accelerometer [14]. The maturing of IMU technology has opened the incorporation of said devices in the civil area thanks to the pairing with conventional (Numerical correlations) and Artificial Intelligence (AI) methods for frequency and time series analysis for its application in SHM. Methods that relate to accelerometers are known as vibration-based methods [14].

2.1 Impact of AI on Infrastructure Monitoring

Conventional methods involve numerical approaches like inverse sensitivity equations or frequency correlation [15] and commonly, these types of methods need handcrafted features and also the manual solving of mathematical equations to make an assessment of the structure's condition; another characteristic of these types of most of the literature is inclined to work on the frequency domain because signals on the frequency domain can be easily interpreted and analyzed. However, it is not limited to just frequency; there are also models in time and time-frequency domains [14, 15].

With the advancements in computer technology and sensing devices, AI has become one of the most sought-after methodologies in recent years, thanks to its data processing and classification capabilities. While conventional approaches from prior years like Natural Frequency Methods, Mode Shape-Based Methods, Mode shape Curvature, and Non-linear Methods, among others, have proved the concepts of IMU technology for structural measurement through hand-crafted features, the recent rise of AI techniques has significantly enhanced SHM [14, 23, 24]. AI techniques have taken great importance in SHM for feature extraction and classification, particularly for applications like damage detection. Methods like Bayesian Networks, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNN), among others, have been applied in SHM, thanks to the capabilities of AI techniques and modern equipment to process the data obtained from the sensors. Figure 3 presents a general overview of conventional and AI techniques used for SHM over the years for vibration methods, highlighting the increasing importance of AI in the field.

With AI, most techniques follow an analysis of frequency or time domain, coming directly from the raw signal of accelerometers or strain gauges; these measurements enable SHM to be treated as a pattern recognition problem. AI algorithms are powerful enough to provide feature extraction and data classification for damage detection, like CNN-based approaches, where a 1D convolution is performed to extract characteristics of the time series data and later provide a classification of the damage detection [21]. AI-driven models have now been used thanks to the capabilities of the techniques to handle a higher input of data and high dimensional data. They are suitable for analyzing complex and non-linear problems because, as for conventional methods, they result in difficulty handling large quantities of data or implementing complex techniques to solve SHM problems. With AI models, the extraction, handling, and learning of data features can be done seamlessly, which

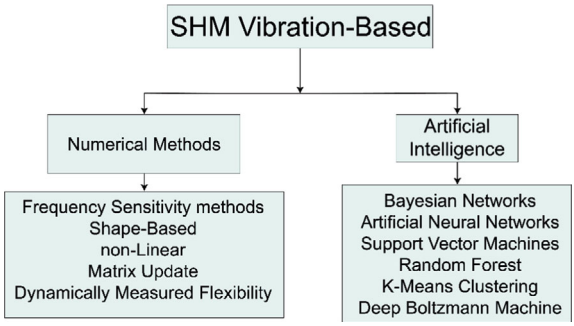


Fig. 3 Structure health monitoring organization diagram with vibration-based methods

provides a broader range of versatility for creating new SHM implementations and provides the construction industry with better, faster, and more robust alternatives for SHM [21, 25].

2.2 Applications of AI in Infrastructure Monitoring

New research in vibration damage detection tends to implement deep AI algorithms to implement feasible and real-time applications for SHM. Models such as ANN, CNN, Bayesian Networks, KNN, and RF are frequently used for detection cases, i.e., M. Betti's et al. research is directly focused on the structural analysis of a three-story steel frame for damage detection using 12 accelerometers to capture the vibration frequency at each intersection point on each level to create a data set with clean and damage data for training an ANN which would provide detection of a structure resonance frequency for a Genetic algorithm training for the damage detection [19]. In Park S.'s work, the authors propose an enhanced vibration-based detector with Bayesian optimization on a 1D CNN for an artificially damaged truss bridge; the data of the bridge comes from eight accelerometers placed on several beams of the structure; this data set is comprised of acceleration reading when the beams are intact, and four other cases where the beam has one of four types of damage defined by the authors of the data set [26]. O. Abdeljaber et al.'s article presents a real-time structural damage detector by implementing 1D CNN, where their application was centered on analyzing real-life buildings, principally modern stadia. Their training data was conceived by developing a grandstand simulator structure with 30 accelerometers to collect undamaged and damaged measurements to detect damage at each structure joint [21]. In S. S. Kourehli's research, a Least Square SVM (LS-SVM) was presented for SHM with an Iterated Improved Reduction System (IIRS) for obtaining the mass and stiffness matrices of the structure for the training of the LS-SVM with simulated (modifying Young's modulus of the elements) and measured modal data on three structures: simply supported beam, a two-story plane frame, and finally in a spring-mass system. The objective of the LS-SVM model was to detect damage to the structure despite the incomplete modal data [22].

In M. Moradi, A. Broer, J. Chiachío, et al., a semi-supervised model is developed to provide a Health Indicator as part of SHM research through the proposition of a Hypothetical HI function to which an NN combined with long-term memory (LSTM) could be trained to learn from the proposed labels obtained from the ideal HI function that analyzes four acoustic emission sensors attached to a stiffener panel [16].

Fernandez-Navamuel et al. research paired a Principal Component Analysis (PCA) feature extractor with a Residual Autoencoder (RA) network for outlier detection in the data measured from the structures; for this research, they used a data set where measurement of strain gauges over several years of two bridges one being the Beltran Bridge in Mexico and the second one is Infante Dom Henrique bridge in Porto [27]. Bao Y. et al. applied SHM for an automatic anomaly damage detector on a bridge; they used 38 accelerometers to obtain multiple samples of data from each sensor to generate a 100×100 pixel image to treat the information as a visual problem [28].

The application of the Vibration-based methods for SHM is vast, this type of damage detection can be implemented on Stadiums, Bridges, and Buildings, among others, because the requirement is to successfully recover data from some vibration sensor like an accelerometer, strain gauge or Acoustic emission sensor for the application of an AI algorithm like ANN, CNN, SVM, PCA-RA, among others.

2.3 Typical Networks in Infrastructure Monitoring

As mentioned previously, some of the standard AI structures to find for data analysis are ANN, SVM, RA, and PCA, among others. These types of networks have proven capable in tasks of pattern recognition and feature analysis, which makes them an excellent alternative to mathematical models. In this section, an overview of some of these networks will be discussed.

2.3.1 ANN

Artificial Neural Networks are comprised of three parts: input, hidden, and output layers, as shown in Fig. 4. ANN can enter the Deep Learning (DL) category when there are multiple hidden layers, and these networks can be trained with raw data but yield better results when working with extracted features from the available data, such as vibration measurements, modal frequency, signal amplitude, and Fourier Transform can be used when training an ANN.

In M. Betti et al.'s work, the recovered time-series data was transformed to the frequency domain to estimate four frequency-dependent indexes, which summarized the structure behavior; these four indexes are (1) probability of tensor modal decomposition, (2) high coherence between DoF, (3) phase close to cero or multiple of pi between DoF, (4) presence of peaks in auto spectral densities. These indexes represent data features used to train and test the ANN through backpropagation.

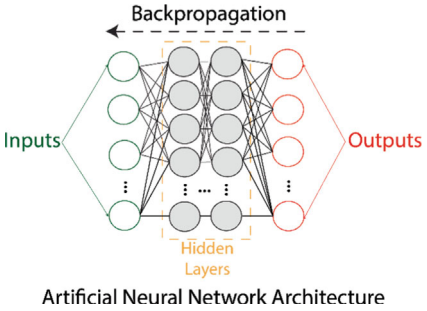


Fig. 4 Typical artificial neural network architecture, where the input, hidden, and output layers are depicted by the colors green, grey, and red, respectively

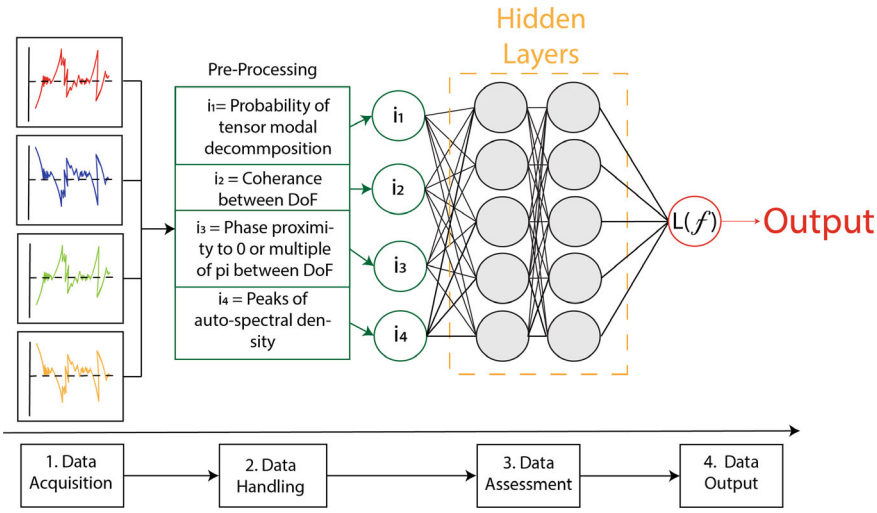


Fig. 5 This figure represents one of M. Betti et al. ANN model. This model comprises of four input neurons, two hidden layers with five neurons each, and one neuron as the output layer for analyzing four frequency-dependent indexes that summarize the structures' behavior

M. Betti et al.'s ANN architecture is shown in Fig. 5, where the streamlined data flow is shown. First, the measured signals are transformed into four indexes representing different frequency behavior characteristics of the analyzed structure. These indexes are then fed to a corresponding neuron in the input layer, followed by two hidden layers with five neurons and a sigmoid activation function for each one. Lastly, the processed information by the first layers passes to the final neuron corresponding to the output layer, which estimates the likelihood $L(f)$ of detecting structural resonance at a specific frequency. The authors also test their model with 10 and 15 neurons in the hidden layer.

2.3.2 SVM

As shown, NN is provided with extracted features to train and test for a specific task; this is also common in several supervised AI algorithms, i.e., in previously mentioned S. S. Kourehli's research, LS-SVM is presented. SVM is used to solve non-linear classification, regression, and pattern recognition tasks, which makes them applicable to SHM through vibration sensors. These types of networks can be viewed as a mapping technique, given that the final result after training is a set of vectors mapped around the original input space of the input data and its corresponding label (x_i, y_i), also known as a hyperplane. The LS-SVM proposed by the author is denoted by Eq. 1.

$$f(x) = \sum_{i=1}^{k} a_i y_i K(x, x_i) + b \tag{1}$$

where (a_i) is the Lagrange multiplier, $K(x, x_i)$ is the Kernel function of the SVM, and b is an error. This equation is obtained by implementing the Lagrange equation and Kurush-Kuhn-Tucker conditions for optimality (KKT).

There are multiple Kernel functions used for SVM, and for this case study, the authors selected a Radial Basis Function (RBF); this function is used because of the non-linear properties; it can be used to help with the mapping of the hyperplane when the input data is not of linear nature, like frequency analysis of a structure in the presence of damage given that it modifies the natural properties without damage. The RBF function is presented in Eq. 2.

$$K(x, x_i) = e^{-\frac{||x - x_i||^2}{2\sigma^2}} \tag{2}$$

where σ is the Bandwidth of the RBF kernel. Overall, Eqs. 1 and 2 are used to relate the input (x_i) to the output (y_i), by minimizing the value margin of Eq. 3 (which includes the least square estimation) and maintaining the equality constraints is shown in Eq. 4.

$$\min j_{(w,e)} = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{i=1}^{k} e_i^2 \tag{3}$$

$$y_i \left(w^T \phi(x_i) + b \right) + e_i = 1, \quad i = 1, \ldots, k \tag{4}$$

where e_i is the slack variable, γ is the regularization parameter, and finally w_i is the weight vector, also known as the hyperplane vector. The w_i's are updated at each iteration of the training process through 10-fold cross-validation and solving the linear equations provided by the KKT conditions of optimality.

The input data x_i of the LS-SVM from S. S. Kourehli's, is computed by 5.

$$\left([K^d_{r,\mathrm{IRS}}] - \lambda^d_{i,r} [M_{r,\mathrm{IRS}}] \right) \{\Phi^d_{i,r}\} = 0, \quad i = 1, 2, \ldots, n \tag{5}$$

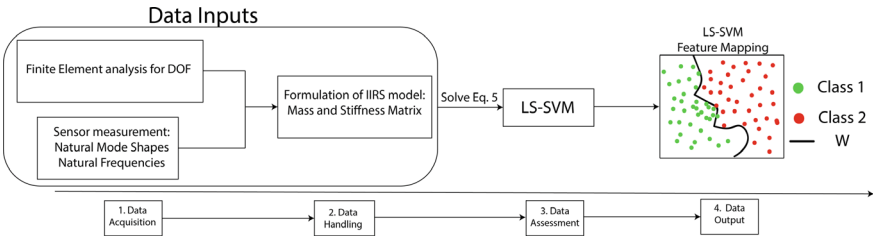


Fig. 6 Graphical representation of the least square support vector machine model from S. S. Kourehli's research

where M and K are the mass and stiffness matrices, and ϕ is the incomplete modal shape of the damaged structure in its reduced state by the IIRS and the number of DOFs used. Figure 6 summarizes S. S. Koureli's algorithms for SHM, where the data inputs are the number of natural mode shapes, natural frequencies, and the DOFs selected.

The test made by the author suggests that the model has the capabilities for its application in SHM, given that with synthetic and measured data, it could identify the damage in the proposed modified section of each structure, for example, in the two-story plane frame, the Mean Square Error in testing was of $7.282 \times e^{-4}$ [22].

2.3.3 CNN

The previous two models discussed for SHM have a similar structure of data flow in their architecture, and comparable models like KNN, RF, and Bayesian Networks, among other ML models, have a specific step for feature extraction of the structure's frequency behavior to then pass to the AI model. In the past decade, DL methods like CNN have received an increased amount of research year after year thanks to their outstanding results for applications like classification, detection, and regression, mainly provided by their capabilities of not requiring feature extraction stage, data management, and versatility for working in one, two or even three dimensions. CNN network includes natively the stage of feature extraction in their streamlined process. A typical 2D CNN architecture comprises a Convolutional layer, an Activation layer, a Pooling layer, and finally, a fully connected layer, also known as an ANN, as shown in Fig. 7.

The Convolutional layer is comprised of the computation of a convolution of an input with a filter bank (also known as kernels) and an activation function which indicates if the result of the convolution will be passed to the following process, the result of the convolution layers are also known as feature maps given that each time a convolution is made with different kernels, different features are obtained. The Pooling layer is used as a dimension reduction, also known as a sub-sampling layer.

Commonly, CNNs are used in computer vision for image tasks like object detection and image classification; they are not limited to working only with images. They

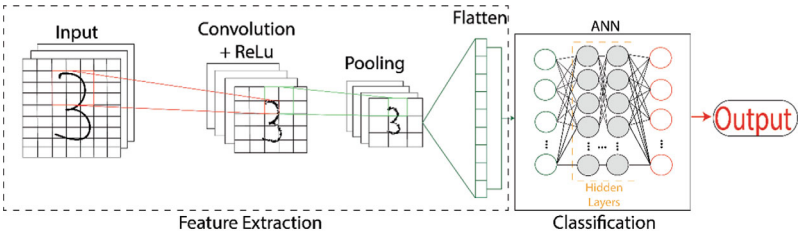


Fig. 7 General overview of a typical CNN architecture

can also be implemented in signal analysis thanks to the properties of the CNN; this type of CNN is known as 1D CNN. In a conventional CNN, a 2D image is used as input and analyzed as a matrix of data, but for 1D inputs, a filter is convoluted to the existing 1D array of data, so the only change made is the dimension of the input, i.e., in the work of O. Abdeljaber et al.'s [21], a 1D CNN is presented for real-time damage detection, utilizing the raw acceleration signal from a test structure (grandstand simulator). The structure is comprised of 30 joints, and at each joint is an accelerometer sensor, which reads the signal when the structure receives a random excitation. And so, for each joint, a 1D CNN is paired to the signal of an accelerometer; the signal comes from a bolt at each joint that, when tightened, is considered undamaged and, when loosened, is considered damaged in the structure. The data inputs n_i are the number of sample frames and are computed by Eq. 6.

$$n_i = \frac{n \times n_T}{n_s} \tag{6}$$

where n_T are the total number of samples in each acceleration signal, and n_s represent the amounts of samples from each frame. O. Abdeljaber et al.'s final network structure is depicted in Fig. 8.

Where the number of frames is $n_i = 2048$ for each joint, and the number of samples from each frame is $n_s = 128$. These samples are then introduced as normalized

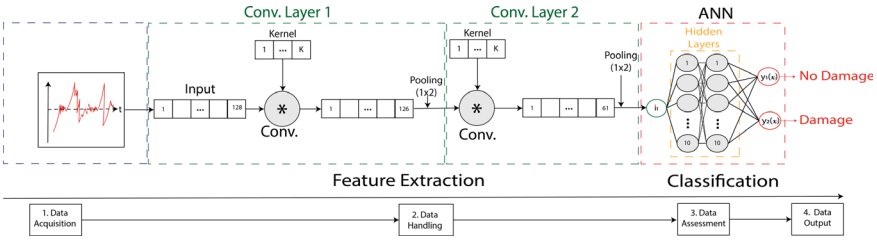


Fig. 8 The architecture of O. Abdeljaber et al.'s 1D CNN for SHM, where the network is comprised of two convolutional layers and an NN with one input neuron, two hidden layers with ten neurons each, and a final output layer with two neurons

data from 1 to –1 to the trained 1D CNN; the kernel size selected was k = 41, and the pooling size is 2; at the end of the structure, ANN receives the extracted features, and with a structure of one input neuron, two hidden layers of ten neurons each and two output neurons provide a classification of the condition of the structure (no damage, damage). The results provided by the authors show the outstanding performance of an AI model capable of damage assessment managing the raw data from the accelerometers, providing a classification error of 0.54% in their proposed scenarios. This work also demonstrates the capabilities of CNN models to work as a powerful tool for researchers in SHM.

3 Applications of IMU Sensors and AI in Infrastructure Monitoring

The implementation of IMU sensors to monitor a structure condition, provides a solution to altern techniques which require a large amount of space or are invasive methods that need to be planned during construction or implement modifications to the structure [29, 30].

It is required to consider that the measurements result from a movement in the structure; hence, if there isn't motion in the system, the instrument will not provide any valuable data for infrastructure monitoring. Buildings, bridges, roads and similar structures are constantly influenced by the vibrations of human steps, noise, and in the case of vehicles or machines, the use of the same and the daily effects of the environment is sufficient for the instrument in an SHM [31–33].

3.1 Structural Health Monitoring Civil Buildings

Some systems are complemented with the aid of an IMU to obtain information about the motion to improve the measurements. In Structural Health Monitoring Systems that implement cameras or lasers, an IMU allows the measurement of the vibrations at which buildings could be subjected and which are outside the camera's Field of View (FOV) [34]. Hence, systems that implement acoustic emissions to determine the noise density and frequency complement their information with an accelerometer, which provides about the vibration.

However, it is important to remember that IMU is subjected to an inherent drift in the measurements, which influence extends through the inertial system, and this error increments through time. Hence, it is required to assist the IMU data with methodologies to mitigate the inherent drift, techniques such as filters, correction algorithms, and machine learning methodologies, provide a solution to mitigate the error, and successfully perform sensor fusion.

3.1.1 Vibrations

Measuring vibrations in buildings aided by an IMU is a common application that uses an accelerometer to observe factors such as the object's natural frequency, resonant frequency, and the building's measurable frequency range. Hence, it is required to consider the accelerometer's sensitivity to the influence of external forces. Even when IMU could be arranged with two or more orthogonal accelerometers, a variation in the frequency measurement is considered noise for the sensor.

As part of this analysis, it is required to determine where the sensors in the building are going to be positioned to obtain significant data properly. This process is called Optimal Sensor Placement (OSP), where the locations are determined iteratively according to the number of available sensors and the degrees of freedom of the building [35].

In Fig. 9 is seen a *Single Degree of Freedom structure* is subjected to a lateral force to determine the highest concentration of energy in the building. Hence, a feature extraction algorithm could be implemented to determine relevant features such as distributions of time of vibrations, localization of sources of vibration, and concentration of energy [36, 37].

Therefore, an artificial intelligence optimization tool can be implemented to proceed with the feature extraction. This can be initiated by a population where the variables possess a velocity and a position vector inside the population. A methodology called "Particle Swarm Optimization" presents a concise algorithm which serves as an example to perform a feature extraction of vibration characteristics of the accelerometer provided data. In this methodology, the defined particles move in a space that is required to find the optimal individual and neighbor positions also called best local *(pbest)* and best global *(gbest)*.

Figure 10 is presented as a flowchart to generalize the feature extraction of significant vibrations in a building. Additionally, it must be considered that to adjust the velocities and positions of the particles, it is required to evaluate with the following Eqs. 7 and 8 [38].

$$v_{ij} = w v_{ij}(t) + c_1 r_1 \left(\text{pbest}_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(\text{gbest}_{ij}(t) - x_{ij}(t)\right) \tag{7}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{8}$$

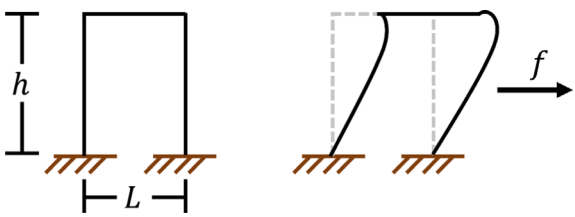


Fig. 9 Single degree of freedom building. As the structure could concentrate vibration in the points where a deformation for vibration occurs, is influenced by a lateral force f

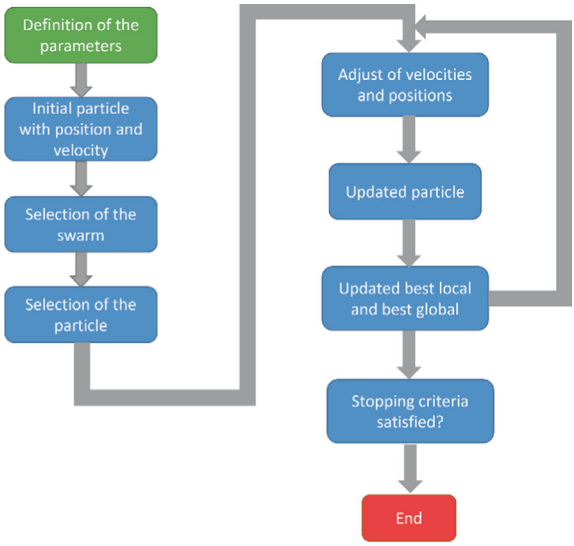


Fig. 10 Particle swarm optimization flowchart, implemented to perform a feature extraction of the vibration obtained by an IMU accelerometer

where $x_{ij}(t)$ and $v_{ij}(t)$ represent the displacement vector and velocity, respectively, for a particle $i$in dimension j during iteration t. A weighted inertia factor w is considered, along with learning coefficients c_1 and c_1. The values r_1 and r_2 are random numbers between 0 and 1 to initialize the process.

This type of methodologies find iteratively new global and optimal solutions to determine the vibrations features when an structure is subjected to seismic motion, human steps, machine vibrations or the influence of external forces. As the data presents information about the resonance in the structure, the detection of displacement or damages in the same, requires implementing additional methodologies to determine the structural condition.

3.1.2 Displacement

Some systems implement IMUs to extract information about displacements in buildings. The techniques take samples of different types of vibration, considering their frequency and duration. Hence, implementing neural network algorithms to assist inertial systems diminishes the error when they measure the displacements in a structure [39].

This network, illustrated in Fig. 11, considers the locations of the structure beams and the forces applied to them to train the network and obtain the displacements [40].

Hence, the subscript i indexes units in the input layer, j the units that are neither inputs or outputs called *"hidden layer"* and w_{ij} denotes the weights in the input to hidden layer [41].

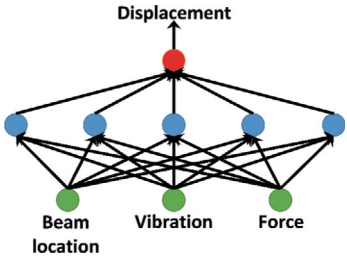


Fig. 11 Input x parameters in a neural network to obtain displacement

An inertial system can also complement other monitoring systems, such as cameras. In the implementation of a camera, the structure of interest places the accelerometers as a reference to the vision system, which requires to use of matching algorithms to align the feature points of the image with the data in the accelerometer [20].

In the presented system (Fig. 12), as a network is implemented to obtain the building boundaries in order to detect displacements, the information is complemented with the IMU's accelerometer information, where the displacements are aligned in the same reference frame that in the camera to simplify calculus. Hence, the accelerometer data is double integrated to obtain displacement in the desired axis. A correlation is implemented between the IMU's calculated displacement and the vision system displacement [42]. The correlation for an *x-axis* and *y-axis* plane is shown in Eq. 9:

$$\rho = \frac{\sum_i \left((x(i) - \mu_x) \times (y(i) - \mu_y)\right)}{\sqrt{\sum_i \left((x(i) - \mu_x)^2 \times (y(i) - \mu_y)^2\right)}} \tag{9}$$

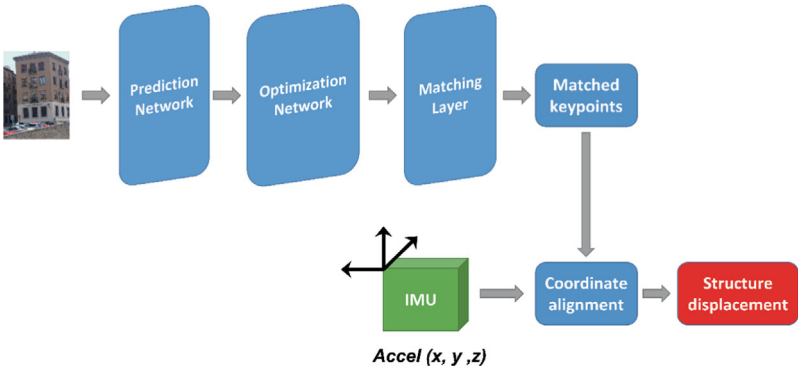


Fig. 12 Alignment of displacement measurements from the IMU accelerometer and vision system for boundary detection in building structures

where $x(i)$ and $y(i)$ are the displacements in the x and y axis, μ_x and μ_y are the mean values of the displacements data.

As previously mentioned, calculating displacement on a building through implementing an IMU requires measurements while the structure is moving. As is possible to identify a variety of failures in a structure, measuring the failures requires complementing the IMU in the inertial system with additional instrumentation.

3.2 Street Mapping and Detecting Irregularities in Road Infrastructure

Well-maintained, paved roads help prevent accidents and reduce pollution by minimizing dust dispersion. They also enhance the driving experience by improving traffic flow and reducing the risk of vehicle damage on uneven or unpaved surfaces. These advantages have driven the development of structural health monitoring (SHM) systems, which aim to continuously assess and monitor the condition of infrastructure, detecting damage or anomalies that may compromise its safety and functionality.

Some SHM applications on roads use images captured by drones to classify irregularities, such as potholes and cracks [43]. However, if the road contains patches or stains, there may be false detections during classification. Lighting is another factor that directly affects image-based SHM systems. For this reason, using inertial sensors (IMU) on roads has been investigated since they do not depend on light or weather conditions. In addition, these sensors are compact and usually non-invasive.

One developed approach uses data from an accelerometer and a gyroscope installed inside a vehicle. The data is processed and fed into a classification algorithm. Two algorithms were used: one based on thresholding and the other on neural networks. The classification is binary: roads in good condition are marked as *"green zone"*, while roads in bad condition (unpaved, potholed, or "crocodile skin") are labeled as *"red zone"* [17].

The author Ugale [17] proposes a system that uses an Arduino microcontroller, an MPU 6050 inertial sensor, a NEO 6M GPS module, and an ESP 8266 chip to enable wifi connectivity. This sensor node is placed on the dashboard of the car (in this case, a Renault Duster). A diagram of the system communication is shown in Fig. 13. The accelerations and angular velocities perceived by the MPU 6050 sensor are sent to the Arduino, which, through the wifi module, transmits the data to a laptop. They are sent to the cloud, processed, and prepared for classification. The binary result is reflected in an interactive map, where the *"red zones"* represent streets in poor condition and the *"green zones"* streets in good condition.

Author Ugale [17] preprocesses the data before feeding it into the classification algorithm to improve accuracy. Where data arrive in packets of six, i.e., each point contains the components *Ax, Ay, Az* (accelerations) and *Gx, Gy, Gz* (angular velocities), corresponding to the raw readings of the MPU 6055 sensor. First, the mean

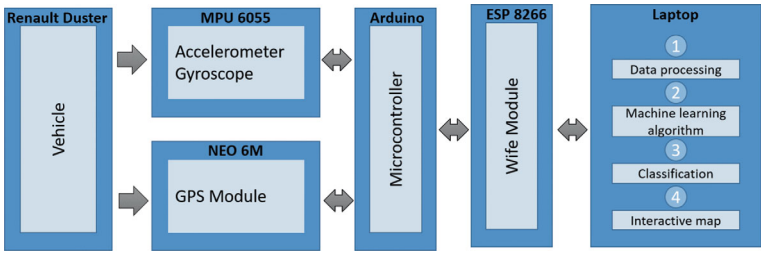


Fig. 13 Diagram of the system proposes by H. Ugale for classifying streets in good and bad condition

acceleration is subtracted from each reading, adjusting only the accelerometer readings so that the accelerations approach zero. A moving average filter is applied to the leveled accelerations and raw gyroscope readings to remove noise. The first derivative of time is then obtained to calculate the rate of change. Finally, each value is squared to ensure all the results are positive.

Once processed, the data can be input into a classification algorithm. The input consists of a vector with six elements (Ax, Ay, Az, Gx, Gy, Gz). For example, Author Ugale [17] used a threshold value selected for each input vector component in a threshold algorithm. The vector is compared to a threshold vector, and each comparison results in a boolean value (0 or 1), thus creating a new vector called V_x. This vector is used to store the value that appears most frequently. If the highest frequency value is 1, the system classifies the point as a road in poor condition*(red zone)*; otherwise, if it is 0, it is a road in good condition *(green zone)* for this case.

Another method mentioned by the author, Ugale [17], is a Neural Network (NN) for classifying good and bad road conditions from the processed IMU sensor data. Figure 14 shows the architecture of the neural network, composed of six layers. The first layer corresponds to the input, which receives the processed readings from the accelerometer *(Ax, Ay, Az)* and the gyroscope *(Gx, Gy, Gz)*. Then there are four hidden layers, and finally, the output layer, which predicts whether the road is in good condition *(green zone)* or bad condition *(red zone)*.

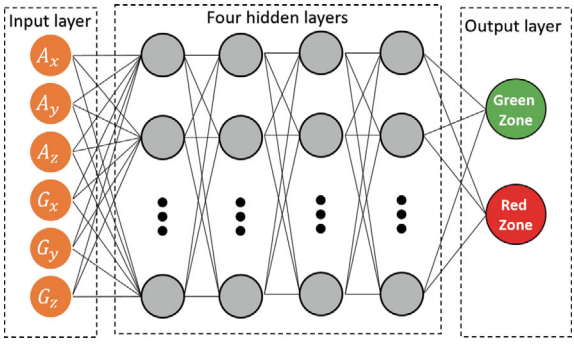


Fig. 14 NN structure for classifying roads in good or bad condition

Two algorithms were mentioned to classify streets in good and bad condition, using the processing of readings from an accelerometer and a gyroscope. The author Ugale [17], obtained satisfactory results but mentioned that the algorithm has some errors in classifying green zones. However, this problem decreases with the NN, concluding that an NN gives better results than a thresholding algorithm for this application. The proposed neural network (NN) algorithm has great potential for the development and improvement of structural health monitoring (SHM) systems.

4 Case of Study in Road Infrastructure Monitoring Using IMU Sensors and AI

This section presents an application for road monitoring using an IMU sensor and an AI algorithm. Unlike similar approaches, our method applies only two operations to the signals before using them as input to the algorithm. By employing the IMU acceleration data as independent variables, we aim to develop a system that classifies speed bumps in real-time, using data windows to train a Convolutional Neural Network (CNN). The data for the CNN will be organized in windows containing the accelerations recorded by the IMU.

As mentioned in Sect. 2.3.3, CNNs are primarily used on images [44]. However, they have proven so efficient that they have started to be used on other types of data, such as signal classification. This section will use a CNN to classify based on signals obtained by an inertial sensor.

4.1 Acquisition and Communication with IMU Sensors

The developed system classifies speed bumps in real time using accelerometer readings from an IMU sensor. A myRIO-1900, a DELL Latitude 7480 laptop, and a Honda Civic Coupe 2012 vehicle were used for the field experiments. A LabVIEW program managed communication with the myRIO, while another LabVIEW program invoked the Python [45] function.

The NI myRIO-1900, a compact device designed by National Instruments, was used for data acquisition. This device offers analog and digital inputs and outputs, audio capabilities, and USB or Wi-Fi connectivity, making it ideal for control applications.

The developed LabVIEW program allowed communication with both inertial sensors through the myRIO. Acceleration readings are displayed in real-time using graphs and indicators and stored in a file for later analysis. Figure 15 shows a program flowchart configuring the I2C communication with the MPU 9255 sensor and simultaneous communication with the myRIO accelerometer. The data is then read in raw format, processed to convert to m/s^2, and displayed in real-time in LabVIEW.

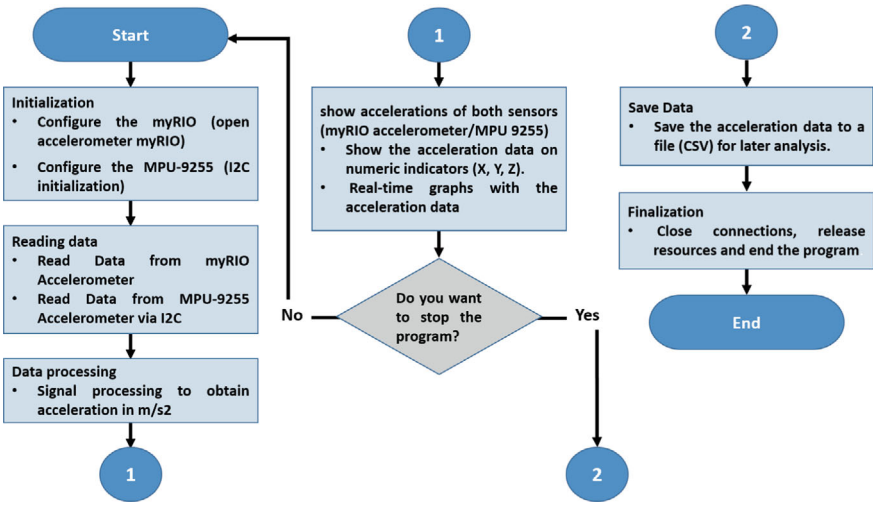

Start
Initialization
• Configure the myRIO (open accelerometer myRIO)
• Configure the MPU-9255 (I2C initialization)
Reading data
• Read Data from myRIO Accelerometer
• Read Data from MPU-9255 Accelerometer via I2C
Data processing
• Signal processing to obtain acceleration in m/s2
1
1
show accelerations of both sensors (myRIO accelerometer/MPU 9255)
• Show the acceleration data on numeric indicators (X, Y, Z).
• Real-time graphs with the acceleration data
Do you want to stop the program?
No
Yes
2
2
Save Data
• Save the acceleration data to a file (CSV) for later analysis.
Finalization
• Close connections, release resources and end the program
End

Fig. 15 Program flowchart in LabVIEW for data acquisition and collection from inertial sensors

Upon completion of the capture, the data is saved in a CSV file, the reading of the sensors is stopped, and the program is closed.

Each IMU sensor follows its communication protocol and signal processing method to convert measurements into accelerations expressed in m/s^2. The NI myRIO-1900 features a three-axis accelerometer with 800 samples per second sampling rate. The accelerations this accelerometer measures are initially expressed in '*g*' units, which are converted to m/s^2 using Eq. 10.

$$\alpha_{ms} = \alpha_g \times 9.80665 \tag{10}$$

where α_{ms} is the acceleration converted into m/s^2 units, α_g is the acceleration obtained from the sensor in '*g*' units, and this measurement is multiplied by the gravity constant approximately *9.80665*.

The MPU 9255, developed by InvenSense, is a nine-axis inertial sensor that integrates an accelerometer, a gyroscope, and a magnetometer. This sensor can read signals and configure parameters through I2C communication, thus facilitating its integration with various microcontrollers, such as Arduino and myRIO.

The proposed program configured the MPU 9255 sensor to read data from the accelerometer. Once communication is established, the sensor provides raw data, where each acceleration measurement is split into two 8-bit values that need to be combined to form a single 16-bit value. For example, The Algorithm 1 describes converting raw accelerometer data from the MPU sensor into acceleration values in physical units. First, two registers are read for each axis *(x, y, z)*: the most significant 8-bit *(MSB)* register, and the least significant 8-bit *(LSB)* register. These two values are combined to form a 16-bit number representing the acceleration in each axis.

This value is then converted into *'g'* units by dividing the result by *16 384*. Finally, the acceleration is expressed in meters per second squared (m/s^2) by multiplying the value in *'g'* by *9.80665*, the acceleration due to Earth's gravity. This procedure is performed for all three axes *(x, y, z)* to obtain their acceleration readings. Below it illustrates these steps.

Algorithm 1 Pseudocode for acceleration estimation from raw data from an MPU sensor

Start

Read raw data from the accelerometer
$X_{MSB} = \text{read}(x_o)$
$X_{LSB} = \text{read}(x_o)$
$Y_{MSB} = \text{read}(y_o)$
$Y_{LSB} = \text{read}(y_o)$
$Z_{MSB} = \text{read}(z_o)$
$Z_{LSB} = \text{read}(z_o)$
Join bytes to form 16-bit values
$X_i = X_{MSB} \times 256 + X_{LSB}$
$Y_i = Y_{MSB} \times 256 + Y_{LSB}$
$Z_i = Z_{MSB} \times 256 + Z_{LSB}$
Convert to "g" units
$X_g = X_i/16384$
$Y_g = Y_i/16384$
$Z_g = Z_i/16384$
Convert to "m/s^2" units
$X_{m/s^2} = X_g \times 9.80665$
$Y_{m/s^2} = Y_g \times 9.80665$
$Z_{m/s^2} = Z_g \times 9.80665$

End

4.2 myRIO Versus MPU 9255 Accelerometer Comparison

Once communication with both sensors was established, tests were performed by leaving the two accelerometers in a static state for several minutes. During this time, acceleration data in m/s^2 was recorded from both sensors to perform a statistical analysis and determine which one offers the best performance for our system. Since the *z-axis* is the most influenced by gravity, the normal distribution of data on this axis was plotted for each sensor. Figure 16 shows that the *z-axis* data distribution of the MPU 9255 sensor is broader than that of the myRIO accelerometer, indicating that the *z-axis* acceleration is more stable on the myRIO compared to the MPU 9255.

Using the measured accelerations on the x, y, and z axes of each sensor, the mean, standard deviation, and root mean square error (RMS) were calculated. The results obtained for each axis are compared in Table 1. The data reveal that the myRIO

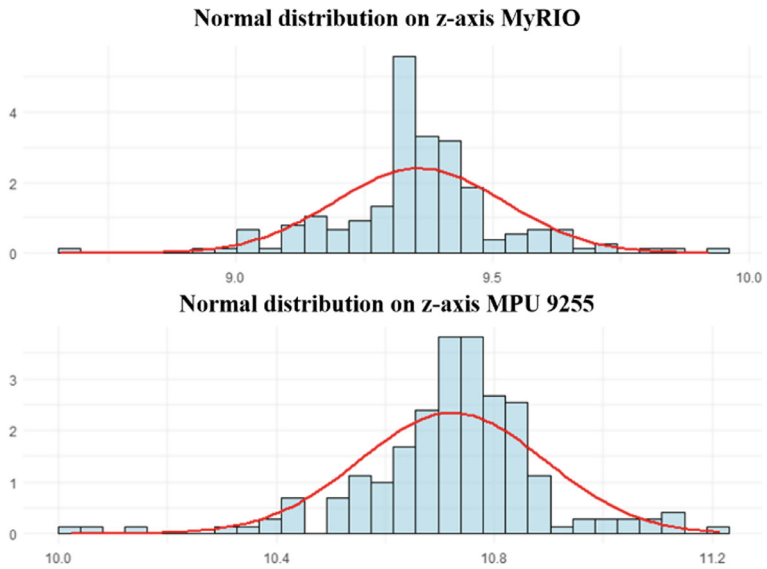


Fig. 16 Normal distribution of *z-axis* data: comparison between the myRIO accelerometer and the MPU 9255

Table 1 Comparison of mean, Standard deviation, and Root mean square error for accelerations measured on the X, Y, and Z Axes of myRIO and MPU 9255 sensors

Statistical measure	myRIO			MPU 9255		
Axis	x	y	z	x	y	z
Mean	0.3531	0.4125	9.2571	0.779	0.0405	10.7229
Standard deviation	0.5857	0.5748	0.166	0.6535	0.5659	0.1706
RMSE	0.0019	0.0019	0.4678	0.003	0.0028	0.6199

accelerometer shows more stability in the static state than the MPU 9255, evidenced by a mean closer to the expected value and a lower standard deviation and root mean square error.

Based on the statistical analysis of the accelerations obtained from both sensors in a static state, we decided to use the myRIO accelerometer for our system.

4.3 *Creating a Dataset to Train a CNN*

In this phase, a CNN network is developed. This programming language is compatible with various environments, such as LabVIEW, which makes it ideal for developing the necessary algorithm. The proposed method is a supervised classification algorithm. Therefore, the dataset is labeled to train the CNN to classify speed bumps on the road. Since there are no pre-existing databases with a dataset with examples of speed bumps and standard surfaces, it is necessary to create one from experimental data. The first step in this process is to define how the inputs will be provided to the CNN.

Since it is complex to identify a speed bump from a single acceleration captured in a single time delta, it is necessary to work with time windows that group several deltas. On average, approximately 145-time deltas are required to capture the entire speed bump, depending on the speed of the speed bump and the vehicle speed. For this reason, it was decided to create windows that include not only the speed bump information but also pre- and post-event data. These windows are organized in a 3D array, composed of 2D arrays, where the accelerations measured in each time delta during the passage over the bump are saved.

After several tests, windows of size 3×308 were defined, where each row represents an axis *(x, y, z)*, and each column corresponds to a time delta. To generate a suitable data set, 65 windows that move sequentially over time were used, capturing the complete behavior of the vehicle as it traverses the speed bump. The initial dataset comprises 756-time deltas organized in three columns *(x, y, z)* containing the measured accelerations. From this data, the 65 windows mentioned above are extracted. This process is illustrated in Fig. 17.

The array of 65 windows contains the data for a single speed bump and is shifted in time to capture the entire passage through the speed bump. An example of a speed bump with 154 time deltas is shown in Fig. 20, using a set of 756 samples (samples 0–755). The first-time deltas of 0 to 300 do not include the speed bump, while samples 301–454 do (154 time deltas). Finally, samples 455–755 do not include the speed bump. From this set of data, shifting windows are created that cover the speed bump. In Fig. 18, the first window (W_1) covers samples 0–307; the second window (W_2)

Fig. 17 The data window generation process for speed bump detection organizes time deltas into 3D windows for CNN training

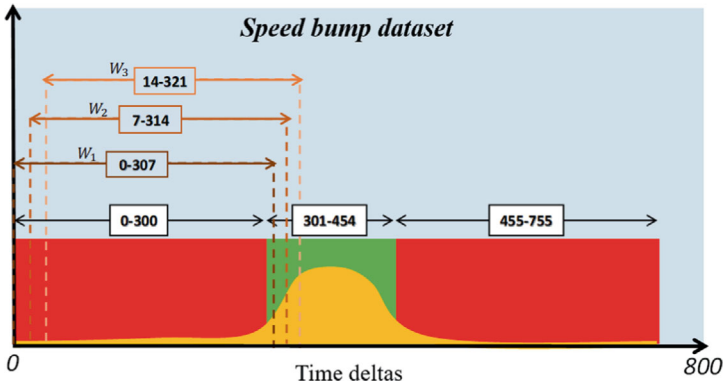


Fig. 18 Shift windows from the speed bump dataset, with a shift of 7-time deltas between each window

Fig. 19 Labels assigned to the 65 windows of a 3D array generated from a speed bump

Window number	first time delta of the window	first time delta of the window
1	0	307
2	7	314
...	...	...
22	147	454
23	154	461
24	161	468
...	...	...
65	448	755

Bump
Nothing

covers samples 7–314; the third (W_3), from 14 to 321, and so on, moving 7-time deltas between each window.

In the example of a speed bump composed of 156-time deltas, a 3D array with 65 windows was generated, whose labels are shown in Fig. 19. The first 22 windows are labeled 'bump' since they contain data from the speed bump's passage. From window 23 onwards, the speed bump has fallen behind and is labeled 'nothing'. Although these windows still contain speed bump data, they correspond to past times. From window 23 to window 65, they are all labeled 'nothing'. This process is repeated for each speed bump in the CNN training set.

4.4 Processing Data

To develop CNN, the data must be pre-processed. Unlike other systems, this process keeps a low computational cost. Each data window fed into the CNN for classification is normalized according to Eq. 11. This normalization converts the accelerations into grayscale data from 0 to 255, where 255 represents the maximum acceleration of $2g$, equivalent to a white pixel in an image. As the acceleration decreases, the value approaches 0, representing shades closer to black.

$$w_l = \frac{W_n}{2g} \times 255 \tag{11}$$

The w_1 is the normalized window, W_n is the current window, the constant $2g$ is the gravity value multiplied by two, and the constant 255 is the maximum value on the grayscale.

4.5 CNN in Python

Once the dataset for training the neural network has been created, a CNN is developed in Python. First, the windows of the 3D array created for training are normalized. Then, the labels are loaded from a CSV file and converted from text *("bump" and "nothing")* to numeric values *(1 and 0)*. Finally, the data is split into training and test sets using the *train_test_split* library, reserving 20% for testing and 80% for training.

The CNN architecture, shown in Fig. 20, includes a 1D convolutional layer, a flattening layer, and two dense layers, the latter containing a *'sigmoid'* activation function for binary classification. The model is compiled with the Adam optimizer and the *'binary_crossentropy'* loss function and trained for forty epochs. Finally, the network weights are saved in a .h5 file.

The CNN is configured with the ReLU activation function (Eq. 12) in all layers except the last one, where he employed the sigmoid function, which is ideal for binary classification problems to perform binary classification using a sigmoid activation function (Eq. 13). This function provides a probability, ranging from 0 to 1, indicating the membership of a window to a specific class, in this case, whether the window

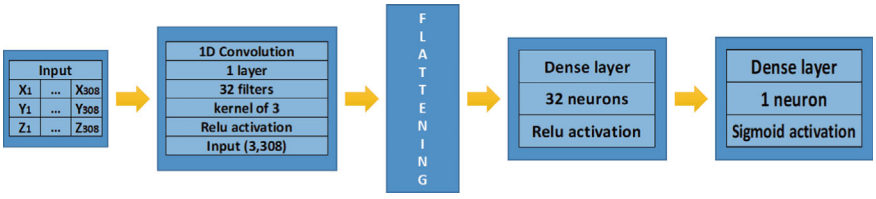


Fig. 20 One-dimensional Convolutional Neural Network (CNN) architecture

corresponds to a *'bump'*.

$$f(x) = \max(0, x) \tag{12}$$

In the ReLU function, if the input x is positive, the output will be x; if x is negative, the output will be 0.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

S(x) represents the output of the sigmoid function for a given input value x. The output value is always in the range of 0–1. The input x is derived from a linear combination of the inputs, including a weighted sum plus a bias.

4.6 Communication Between Python and LabVIEW

Once the neural network for classification was configured, a LabVIEW program was developed that invokes a Python function containing the CNN. In real-time, the program sends windows of data to the CNN and receives the percentage probability that those data correspond to a speed bump determined by the sigmoid function. The threshold in LabVIEW was set to 0.8, meaning that if the probability of a window being a *"bump"* is equal to or greater than 80%, the system classifies it as such. This condition is illustrated in Eq. 14:

$$p[S(x)] = \begin{cases} 1, & \text{if } S(x) \geq 0.8 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where *p[S(x)]* is the boolean value to indicate whether it is a speed bump, *S(x)* represents the value obtained by the sigmoid function.

To invoke the CNN, a Python function must be created that receives a normalized data window, loads the trained CNN model, and returns a probability between 0 and 1, indicating the probability that the window corresponds to a speed bump. The Python function is given the path to the file containing the function, its name, the type of return data (the probability), and the data window that serves as input. The window sent as input has a 3×308 format and is updated every ten milliseconds with real-time data. The function returns the probability that the window is a speed bump.

In LabVIEW, a condition is programmed based on the value obtained from the prediction. If the value is greater than or equal to 0.8, a light will turn on in the user interface, indicating that it is a *"bump"* If the value is less than 0.8, the light will turn off, indicating it is *"nothing"*

The developed programs must communicate in parallel for our system to work in real-time. The process starts with the LabVIEW program, which connects to the myRIO to read the accelerometer data in real-time. Using a producer-consumer

scheme, the data is stored in a queue and converted to accelerations in units of m/s^2. This data is then sent to the second LabVIEW program through a shared variable, which communicates with Python to perform the classification. In the second LabVIEW program, a producer-consumer structure receives and stores the accelerations in a queue. The data is then organized into three windows of 308, normalized, and sent to Python for classification. The Python function predicts whether the data window belongs to the *"bump"* class displayed in the LabVIEW program interface.

4.7 Experimentation and Results

When training the one-dimensional CNN, an accuracy of 91.03% and a loss of 0.01851 were obtained. The loss is shown in Fig. 21.

With successful communication between the programs, real-time experimentation can begin. For testing, the following equipment was used: a laptop, a myRIO-1900, an inverter to power the microcontroller, and a 2012 Honda Civic.

Several experiments were conducted to evaluate the system's effectiveness, considering that road irregularities can influence its performance. Two thresholds were defined for speed bump classification: 0.8 and 0.9. During the real-field experiment, 20 different speed bumps were passed through. With the 0.8 threshold, 16-speed bumps were correctly classified; by raising the threshold to 0.9, the number of correct classifications increased to 18, resulting in 18 true positives and two false negatives. These results demonstrate the effectiveness of the system in detecting speed

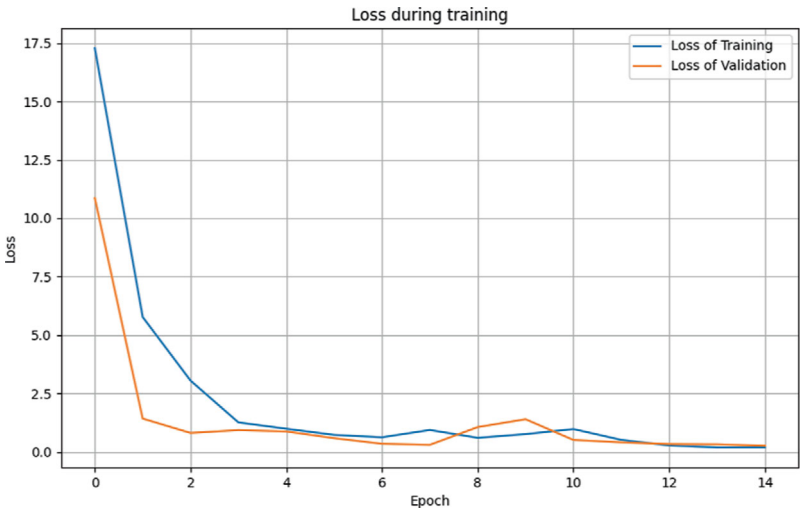


Fig. 21 Loss graph using training and test data

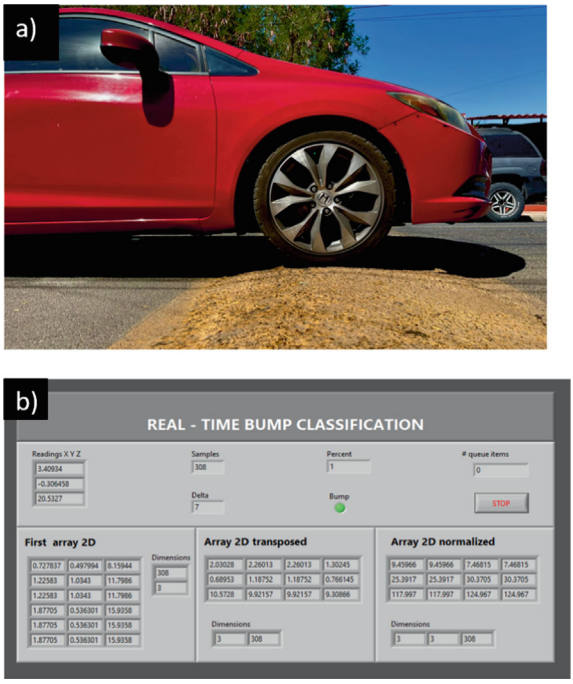


Fig. 22 Experiments and real-time classification: **a** Photograph of the car passing the bump. **b** Screenshot of the system working in real-time classifying the bump in the image **a** as a *"bump"*

Table 2 Results obtained in speed bump classification

Real-time speed bumps detection		
Threshold	0.8	0.9
True	16	18
False	4	2
Accuracy (%)	80	90

bumps. However, errors were observed during driving, as external disturbances generated false positives. These disturbances can be attributed to road conditions (cracks, potholes, stones) or vehicle speed. Tests were performed at a maximum speed of 60 km/h on a flat road, while when passing over speed bumps, the speed was kept below 30 km/h. Figure 22 illustrates two images: image (a) shows a photograph of the vehicle passing over a speed bump, and image (b) captures the system in real-time, classifying the situation in the image (a) as a "speed bump".

In conclusion, the experiment's results show that using a threshold of 0.8, an accuracy of 80% was achieved, while with a threshold of 0.9, the accuracy increased to 90%, as shown in Table 2.

5 Conclusion

Integrating artificial intelligence and compact IMU sensors has transformed structural health monitoring systems, achieving accurate and noninvasive monitoring. This chapter explores how the advancement in MEMS technology has enabled IMUs, due to their reduced size and weight, to be used in diverse applications, particularly in SHM, where AI has further expanded its functionality.

Previously, SHM systems relied on numerical methods that required constant monitoring and yielded subjective results. With the implementation of algorithms such as SVM, ANN, and CNN, these systems have become more robust and accurate. For example, neural networks can detect structural resonance at specific frequencies [19], monitor displacements [20], and assess the condition of roads [17]. On the other hand, SVM models facilitate structural damage detection even with incomplete modal data [22]. A CNN-based system was implemented using accelerometer data, which has proven effective for bump detection, achieving an accuracy of over 90%. IMUs typically include accelerometers, gyroscopes, and magnetometers, where their readings require processing to be useful. Challenges such as cumulative deviation due to bias persist, making applying advanced algorithms or filters necessary. In the future, reducing this cumulative deviation will remain a priority. Applications of IMUs and AI in structural health monitoring continue to expand, encompassing structural integrity monitoring of buildings and roads and identifying irregularities in roadways. These technologies are consolidating as key tools for safe and efficient structural monitoring in civil engineering. The synergy between IMUs and AI promises to revolutionize SHM, paving the way to a safer and more reliable future.

Acknowledgements Universidad Autónoma de Baja California, México, and CONAHCYT, Mexico supported this chapter.

References

1. G. G. Samatas and T. P. Pachidis, “Inertial measurement units (IMUs) in mobile robots over the last five years: A review,” *Designs*, vol. 6, no. 1, p. 17, 2022.
2. H. Zhao and Z. Wang, “Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion,” *IEEE Sens. J.*, vol. 12, no. 5, pp. 943–953, 2012.
3. P. V. Sreevidya, J. Khan, H. C. Barshilia, C. M. Ananda, and P. Chowdhury, “Development of two axes magnetometer for navigation applications,” *J. Magn. Magn. Mater.*, vol. 448, pp. 298–302, 2018.
4. N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (IMU) sensor applications,” *Int. J. Sign. Process. Syst.*, pp. 256–262, 2013.
5. Z. You, *Space Microsystems and micro/Nano Satellites*. Oxford, England: Butterworth-Heinemann, 2017.
6. W. H. Ko, “Trends and frontiers of MEMS,” *Sens. Actuators A Phys.*, vol. 136, no. 1, pp. 62–67, 2007.

7. M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4712–4719.
8. C. Cheng, X. Li, L. Xie, and L. Li, "A unmanned aerial vehicle (UAV)/unmanned ground vehicle (UGV) dynamic autonomous docking scheme in GPS-denied environments," *Drones*, vol. 7, no. 10, p. 613, 2023.
9. S. Manish, A. Akash, S. Rupika, and K. Lakshmi, "Wearable audio and IMU based shot detection in racquet sports," 2018.
10. P. Nath and A. Malepati, "IMU based accident detection and intimation system," in *2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*. IEEE, 2018, pp. 1–4.
11. R. R. Minor and D. W. Rowe, "Utilization of GPS/MEMS-IMU for measurement of dynamics for range testing of missiles and rockets," in *IEEE 1998 Position Location and Navigation Symposium (Cat. No.98CH36153)*. IEEE, 2002, pp. 602–607.
12. C. Chen, Z. Yanshun, and L. Chunyu, "Surveying method of measurement while drilling based on the inertial sensor," in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*, vol. 15. IEEE, 2010, pp. 1192–1195.
13. M. E. Tanner and J. M. Protz, "Design of a nano-imu with chemical recording," in *Information Sciences 2007*. WORLD SCIENTIFIC, 2007, pp. 137–143.
14. R. Hou and Y. Xia, "Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019," *J. Sound Vib.*, vol. 491, no. 115741, p. 115741, 2021.
15. E. P. Carden and P. Fanning, "Vibration based condition monitoring: A review," *Struct. Health Monit.*, vol. 3, no. 4, pp. 355–377, 2004.
16. M. Moradi, A. Broer, J. Chiachío, R. Benedictus, T. H. Loutas, and D. Zarouchas, "Intelligent health indicator construction for prognostics of composite structures utilizing a semi-supervised deep neural network and SHM data," *Eng. Appl. Artif. Intell.*, vol. 117, no. 105502, p. 105502, 2023.
17. H. Ugale, P. Patil, S. Chauhan, and N. Rao, "IoT system for sensing condition of roads using IMU sensors," in *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. IEEE, 2021, pp. 344–349.
18. A. Güemes, "Shm technologies and applications in aircraft structures," in *Proceedings of the 5th International Symposium on NDT in Aerospace, Singapore*, vol. 1315, 2013, p. 4.
19. M. Betti, L. Facchini, and P. Biagini, "Damage detection on a three-storey steel frame using artificial neural networks and genetic algorithms," *Meccanica*, vol. 50, no. 3, pp. 875–886, 2015.
20. J. Shan, P. Huang, C. N. Loong, and M. Liu, "Rapid full-field deformation measurements of tall buildings using UAV videos and deep learning," *Eng. Struct.*, vol. 305, no. 117741, p. 117741, 2024.
21. O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *J. Sound Vib.*, vol. 388, pp. 154–170, 2017.
22. S. S. Kourehli, "LS-SVM regression for structural damage diagnosis using the iterated improved reduction system," *Int. J. Struct. Stab. Dyn.*, vol. 16, no. 06, p. 1550018, 2016.
23. B. Moaveni, A. Stavridis, G. Lombaert, J. P. Conte, and P. B. Shing, "Finite-element model updating for assessment of progressive damage in a 3-story infilled RC frame," *J. Struct. Eng. (N. Y.)*, vol. 139, no. 10, pp. 1665–1674, 2013.
24. H.-P. Chen and N. Bicanic, "Assessment of damage in continuum structures based on incomplete modal information," *Comput. Struct.*, vol. 74, no. 5, pp. 559–570, 2000.
25. O. Avci, O. Abdeljaber, and S. Kiranyaz, "An overview of deep learning methods used in vibration-based damage detection in civil engineering," in *Conference Proceedings of the Society for Experimental Mechanics Series*. Cham: Springer International Publishing, 2022, pp. 93–98.

26. S. Park and S. Kim, "Enhancing vibration-based damage assessment with 1D-CNN: Parametric studies and field applications," *KSCE J. Civ. Eng.*, vol. 28, no. 7, pp. 2934–2951, 2024.
27. A. Fernandez-Navamuel, F. Magalhães, D. Zamora-Sánchez, Á. J. Omella, D. Garcia-Sanchez, and D. Pardo, "Deep learning enhanced principal component analysis for structural health monitoring," *Struct. Health Monit.*, vol. 21, no. 4, pp. 1710–1722, 2022.
28. Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Struct. Health Monit.*, vol. 18, no. 2, pp. 401–421, 2019.
29. F. J. Pallarés, M. Betti, G. Bartoli, and L. Pallarés, "Structural health monitoring (SHM) and nondestructive testing (NDT) of slender masonry structures: A practical review," *Constr. Build. Mater.*, vol. 297, no. 123768, p. 123768, 2021.
30. J. C. Rodríguez-Quiñonez, J. A. Valdez-Rodríguez, M. J. Castro-Toscano, W. Flores-Fuentes, and O. Sergiyenko, "Inertial methodology for the monitoring of structures in motion caused by seismic vibrations," *Infrastructures*, vol. 9, no. 7, p. 116, 2024.
31. C. Zhang, A. A. Mousavi, S. F. Masri, G. Gholipour, K. Yan, and X. Li, "Vibration feature extraction using signal processing techniques for structural health monitoring: A review," *Mech. Syst. Signal Process.*, vol. 177, no. 109175, p. 109175, 2022.
32. M. Mishra, P. B. Lourenço, and G. V. Ramana, "Structural health monitoring of civil engineering structures by using the internet of things: A review," *J. Build. Eng.*, vol. 48, no. 103954, p. 103954, 2022.
33. G. Fan, J. Li, and H. Hao, "Vibration signal denoising for structural health monitoring by residual convolutional neural networks," *Measurement (Lond.)*, vol. 157, no. 107651, p. 107651, 2020.
34. K. S. Hahm and B. W. Anthony, "Machine learning-based gait health monitoring for multi-occupant smart homes," *Internet of Things*, vol. 26, no. 101154, p. 101154, 2024.
35. Y. S. Erdogan and M. Ada, "A computer-vision based vibration transducer scheme for structural health monitoring applications," *Smart Mater. Struct.*, vol. 29, no. 8, p. 085007, 2020.
36. B. Li, D. Li, X. Zhao, and J. Ou, "Optimal sensor placement in health monitoring of suspension bridge," *Sci. China Technol. Sci.*, vol. 55, no. 7, pp. 2039–2047, 2012.
37. J. Humar, *Dynamics of structures*. Boca Raton, FL: CRC Press, 2012.
38. D. Arezzo, S. Quarchioni, V. Nicoletti, S. Carbonari, F. Gara, C. Leonardo, and G. Leoni, "SHM of historical buildings: The case study of santa maria in via church in camerino (italy)," *Procedia Struct. Integr.*, vol. 44, pp. 2098–2105, 2023.
39. S. Kavitha, R. Joseph Daniel, and K. Sumangala, "Design and analysis of MEMS comb drive capacitive accelerometer for SHM and seismic applications," *Measurement (Lond.)*, vol. 93, pp. 327–339, 2016.
40. A. Sivasuriyan and D. S. Vijayan, "Prediction of displacement in reinforced concrete based on artificial neural networks using sensors," *Measur. Sens.*, vol. 27, no. 100764, p. 100764, 2023.
41. S. A. Wegner, "Neural networks," in *Mathematical Introduction to Data Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2024, pp. 225–260.
42. T. Huang, C. Yuan, P. Li, S. Xu, and Q. Kong, "Spatial displacement tracking of vibrating structure using multiple feature points assisted binocular visual reconstruction," *Eng. Appl. Artif. Intell.*, vol. 126, no. 106889, p. 106889, 2023.
43. I. A. S. Galeano and J. A. H. Maldonado, "Revista AZIMUT," *Azimut*, vol. 10, no. 1, 2019.
44. D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics (Basel)*, vol. 10, no. 20, p. 2470, 2021.
45. A. Azevedo, *Python para todos: Iniciante*. Independently Published, 2021.

SDG 11 Sustainable Cities and Communities

Analysis of Geographic Coordinates Initialization Methods for Unmanned Aerial Vehicles Localization

Gabriela Rodríguez-Cortés, Anabel Martínez-Vargas, MA Cosío-León, and Oscar H. Montiel-Ross

Abstract How a metaheuristic population is initialized significantly impacts its final performance. In this chapter, an analysis of three population initialization methods within an evolution strategy is conducted. This metaheuristic is employed to localize unmanned aerial vehicles (UAVs) in a disaster-stricken area to establish a temporary communication network. The UAVs must be positioned to maximize the probability of successful voice services (PSVS). According to the findings, our proposed strategy of localizing UAVs within a high-density area yields solutions with a PSVS above 0.77, which translates to a 77% probability of a successful network connection, ensuring reliable communication in emergency scenarios.

Keywords Initialization method · Metaheuristics · Optimization of UAV locations · Evolution strategy

G. Rodríguez-Cortés
Doctorado en Ciencias y Tecnologías Avanzadas, Universidad Politécnica de Pachuca, Zempoala, Hidalgo, México
e-mail: lizeth.rodriguez.cortes@gmail.com

A. Martínez-Vargas (✉) · MA Cosío-León
Dirección de Investigación, Innovación y Posgrado, Universidad Politécnica de Pachuca, Zempoala, Hidalgo, México
e-mail: anabel.martinez@upp.edu.mx

MA Cosío-León
e-mail: ma.cosio.leon@upp.edu.mx

O. H. Montiel-Ross
Centro de Investigación y Desarrollo de Tecnología Digital, Instituto Politécnico Nacional, Tijuana, Baja California, México
e-mail: oross@ipn.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_25

1 Introduction

The precise localization of unmanned aerial vehicles (UAVs) is a critical challenge, as it involves surveillance, environmental monitoring, search and rescue missions, and communication emergencies. Metaheuristic algorithms are frequently employed to tackle these kinds of optimization issues. These algorithms require initialization, which is the first phase of searching for one good solution to the problem. Initialization methods directly influence the effectiveness of metaheuristics employed to tackle complex optimization problems [1]. The initialization method sets the stage for the search process. An efficient initialization strategy is crucial, as it directly influences the convergence rate and quality of the solutions produced by the metaheuristic algorithms. However, traditional methods like random initialization can occasionally suffer from premature convergence and suboptimal results due to their inability to efficiently use search space information [2].

Several works have reported their findings on initialization methods. For example, work in [1] presents a study focusing on local search and generic initialization methods, where detailed knowledge about the problem is not required. In contrast, the authors in [3] examined the initialization phase of five algorithms: differential evolution (DE), particle swarm optimization (PSO), cuckoo search (CS), artificial bee colony (ABC), and genetic algorithm (GA). They highlighted that some algorithms are more sensitive to initialization than others. Their experiments were made with various test functions with different properties and modalities, showing significant differences between the initialization methods. In Addition, the authors emphasized the importance of studying the quality and diversity of solutions and their impact on algorithm performances. Inverse optimization is another initialization method often used to infer constraints or unknown problem parameters, as demonstrated in [4]. They used it to deduce linearly feasible solutions and to establish a starting point for the first screening of potential future solutions according to their viability.

In these works, the experimentation was performed on known problems and known test instances and functions, but there is limited practical guidance on effective initialization techniques. In some cases, researchers have developed unique problem-specific initialization methods. For example, in [5] a special and random sampling evolutionary algorithm was used to estimate the parameters of digital integrated circuits. Node degrees were employed in [6] to initialize a network disintegration problem. These specialized methods, while effective, can be complex to implement and computationally demanding.

This work presents a comparative analysis of geographic coordinate initialization methods for strategic UAV localization to provide communications in disaster situations. We compare three strategies: the classic random method, the random request mobile user (RMU) method, and the density RMU method. The main goal is to identify the initialization method that best balances exploration and exploitation to optimize the performance of the $(1 + 1)$-Evolution Strategy ($(1 + 1)$-ES). $(1 + 1)$-ES has demonstrated effectively solving optimization problems. The main contributions of this work are as follows:

- We carry out an analysis to compare three geographic coordinate initialization methods to optimize UAV localization for communication support in disaster scenarios.
- We evaluate the performance of the three methods (the classic random method, the random RMU method, and the density RMU method) to determine their effectiveness in balancing exploration and exploitation during the search process.
- We demonstrate that the proposed initialization methods significantly influence the performance of the (1 + 1)-ES) for UAV localization, leading to solutions with a higher probability of successful voice services (PSVS) in emergency scenarios.

The rest of the work is organized as follows: Sect. 2 discusses the background of the (1 + 1)-ES and its application in solving the UAV localization problem. Section 3 describes the three initialization methods evaluated in this study. Section 4 presents the experimental results and compares the performance of each method. Finally, in Sect. 5, we write the conclusion and any future research directions.

2 Background

This section will provide a preamble on the background (1 + 1)-ES algorithm and how it solves the location UAV problem. We begin by showing the basic principles of the (1 + 1)-ES algorithm. After, we outline the approach taken to apply the (1 + 1)-ES algorithm to resolve this problem.

2.1 Basic Principles of (1 + 1)-ES

(1 + 1)-ES is an evolutionary algorithm mainly applied to solve optimization problems in continuous search spaces. It is inspired by biological evolution dating back to the 1960s [7] and uses a one-fifth success rule for the mutation of the individuals [8]. The (1 + 1) is the algorithm's configuration; one offspring is generated from a single parent in each generation, imitating the asexual reproduction mechanism of biological evolution. Some of its characteristics are:

1. To produce quality solutions with minimal evaluations of the objective function using a small population size.
2. To operate on the natural representation of the problem. Therefore, it does not use a genotype-phenotype mapping for the object parameters.
3. The set of decision variables that make up the individual includes the parameters that influence the mutation operator too.
4. It employs mutation as a variation operator.

The (1 + 1)-ES algorithm operates under a continuous Gaussian mutation that allows larger mutations to generate, thus escaping local optima. The mutation follows

a Gaussian distribution with a zero mean and standard deviation (σ) [9]. σ is the mutation step size. The one-fifth success rule is used for online adjustment of the step size [10]. Online self-adaptation of the step size varies during the execution algorithm, i.e., the step size controls the mutation. It defines the magnitude of the change in the solution during the algorithm's evolution.

Additionally, the (1 + 1)-ES algorithm can handle uncertainty, ensuring that the generated solutions are both effective and efficient. The simplicity of (1 + 1)-ES makes it attractive for the location UAV problem, where a balance between computational efficiency and effectiveness in finding solutions is required. Furthermore, its iterative nature and ability to explore the solution space efficiently make it a powerful tool for adjusting the placement of UAVs in emergency scenarios.

2.2 *Solving Location UAVs Problem with (1 + 1)-ES*

To deploy the UAV in locations with the highest PSVS, we consider a simulation scenario where communication is disrupted due to a natural disaster. The affected area is square-shaped L kilometers per side. Geographically, the area is limited by $[x_{\min}, x_{\max}]$ for the latitude, and $[y_{\min}, y_{\max}]$ for the longitude. We assume that emergency services have several UAVs to deploy and to build a temporary cellular network in the affected area. UAV has technological capabilities and functionalities that allow it to operate as eNodeB, which is suitable for transmission voice services. Figure 1 shows the simulation scenario of this work.

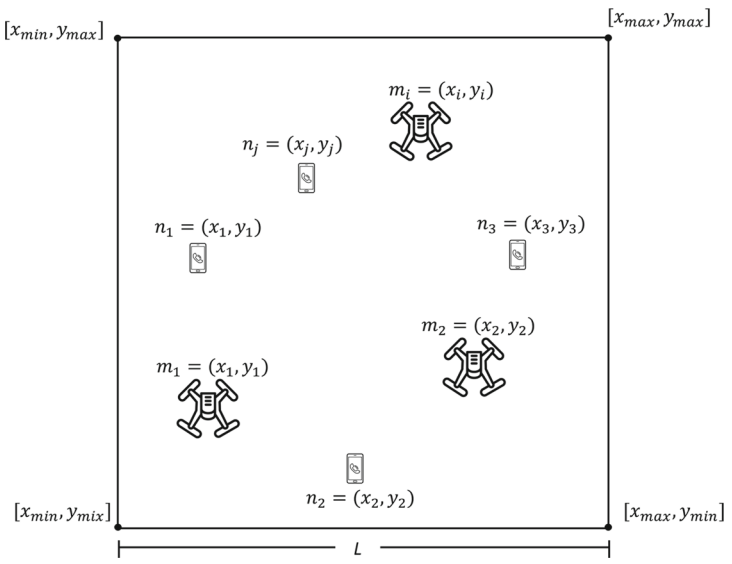


Fig. 1 Simulation scenario for deploying UAVs in optimized positions that maximize PSVS

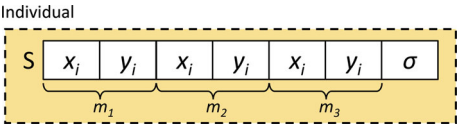


Fig. 2 Example of an individual for (1 + 1)-ES algorithm

The (1 + 1)-ES algorithm evaluates the proposed simulation scenario. An individual for the algorithm is a solution to the problem such as in Fig. 2, individual S includes one σ value along with the latitude and longitude coordinates of each available UAV. This figure represents the solution with three UAVs, detonated by m_1, m_2, and m_3.

To begin the evaluation of the (1 + 1)-ES algorithm, we simulate capturing the state of the network (snapshot) during a specific time, i.e., loading the geographic locations of all present mobile users (MUs) and randomly selecting RMUs (users that start one call inside the affected area). During the evaluation of each snapshot, the number and positions of the MUs remain constant, but the number of RMUs changes according to the dynamics of the Erlang traffic model. This created snapshot is taken by (1 + 1)-ES for processing the according to Fig. 3.

First, from line 1 to line 3, an archive of successful mutations A is created, followed by the iteration counter variable t and the success rate p_s, which are set to zero. The individual S is initialized in line 4, setting the initial value of σ and generating the geographic locations of the UAV. Here, the geographic locations of the UAVs are generated through the initialization methods described in Sect. 3. In line 5, S is set how the parent S_p. In line 6, the fitness value is computed through the objective function. The objective function indicates the solution quality, i.e., the offered PSVS by the set of geographic locations of the available UAV. Its value is stored in Fx_p. From lines 7 to 24, the evolution loop is applied. In line 8, the iteration counter variable increases in one unit. Subsequently, In line 9, a multivariate normal distribution $\mathcal{N}(0, I)$ with a mean vector of 0 and a variance-covariance matrix is added to S_p to create the offspring S_o. In line 10, S_o is evaluated in the objective function, such as the parent S_p, and its fitness is stored in Fx_o. The selection of the fittest individual is solved from line 11 to line 17. A success case is reached if the offspring fitness value Fx_o is greater than the parent fitness value Fx_p. As a result, in the following iteration, the child S_o becomes the parent S_p. In this instance, the archive A updates the $\#successes$ variable in one unit. If the offspring fitness value Fx_o is lower than the parent fitness value Fx_p, a failure case arises, retaining the parent S_p for the following iteration. In this case, $\#failures$ variable increases one unit to its counter. From line 18 to line 22, the 1/5-success rule for step size adaptation σ is used. To do so, the step size σ is adjusted over a predetermined number of iterations z. According to the 1/5-success rule, the step size σ is decreased if the success rate p_s is less than 1/5, but it remains unchanged if the success rate p_s is equal to 1/5. Conversely, if p_s increases above 1/5, the step size σ is increased. Finally, the step size σ is updated in the line 23. The evolution process of the (1 + 1)-ES finishes when the number of iterations $T_{\max}$ is reached. The fittest individual is the one who obtains the maximum achieved by the algorithm.

Data: The individual S, the step-size initial σ, the step size adaptation z, the stop condition T_{max}
Result: The best individual S

Initialize archive A for storing successful mutations;
$p_s = 0$;
$t = 0$;
Initialize individual S;
$S_p = S$;
$Fx_p = P(S_p)$;
while $t < T_{max}$ **do**
  $t = t + 1$;
  $S_o = S_p + \sigma \cdot \mathcal{N}(0, I)$; /* · is a dot product */
  $Fx_o = P(S_o)$;
  if $Fx_o > Fx_p$ **then**
    $S_p = S_o$;
    $Fx_p = Fx_o$;
    store success in A;
  else
    store failure in A;
  end
  if $t \mod z = 0$ **then**
    get $\#successes$ and $\#failures$ from at most $10z$ entries in A;
    $p_s = \frac{\#successes}{\#successes + \#failures}$;
    $\sigma' = \begin{cases} \sigma * \mathrm{c} & \text{if } p_s < 1/5 \\ \sigma \div \mathrm{c} & \text{if } p_s > 1/5 \\ \sigma & \text{if } p_s = 1/5 \end{cases}$;
  end
  $\sigma = \sigma'$;
end

Fig. 3 (1 + 1)-ES for UAV locations problem

3 Initialization Methods

Although the evolutionary strategy is a population-based algorithm, meaning it maintains and improves multiple candidate solutions, the configuration used in this paper (1 + 1) is based on a single solution. This implies that the algorithm focuses on modifying and improving one candidate solution, which is iteratively refined during the algorithm's evolution. This means that the initial diversity comes exclusively from the distribution of the generated values of the single candidate solution and thus can lead to limited exploration if an individual sufficiently representative of the search space is not generated. Therefore, sufficient care must be taken when proposing an initialization method.

In this paper, the (1 + 1)-ES algorithm configuration requires the initialization of only one solution. Therefore, we design three UAVs-ABS geographic coordinate initialization methods to evaluate (1 + 1)-ES and obtain its best performance. They are described below.

3.1 Classic Random Method

The classic random method follows the traditional approximation for many metaheuristic algorithms. This method uses basic random number generators. This is available without any limitations on the size of the population or the dimensions in a variety of programming languages. This method randomly selects a subset of individuals or solutions from a large set of individuals (a population). Hence, each individual is equally likely to be chosen during sampling. This causes unbiased sampling.

In this work, the classic initialization method allows each geographic coordinate to have the same occurrence probability. That means that UAV's initial geographic coordinates are likely to be located in an area without the request of mobile users (RMUs), i.e., the $(1 + 1)$-ES algorithm begins its search process in spaces where there is a probability that voice services are not needed. This could be a disadvantage in the context of this work since the location of the UAVs is supposed to be provided in the shortest time due to the situation's urgency.

3.2 Random Request Mobile Users Method

In the random RMU method, the geographic coordinates of the UAV take geographic coordinates at random from among the users requesting services (RMU). This assignment follows a uniform random distribution. While the random RMU method runs the risk of falling into suboptimal areas due to the limited exploration capacity in the early stages of the algorithm, the method is simple and efficient in terms of computational resources. This is an advantage given the nature of the problem, as time is a critical factor in ensuring that services are provided efficiently and on time. In addition, it allows the algorithm's evolution process to begin in areas where it is known that there are users needing services. This means that $(1 + 1)$-ES begins its search in areas with attendance of RMUs.

3.3 Density Request Mobile Users Method

In contrast to the random RMU method, the density-based RMU method prioritizes regions with a high concentration of RMUs during initialization. This process assigns the geographic coordinates of RMUs to UAV, provided the RMU is situated within a high-density area. By ensuring that the search begins in these critical regions, the method increases the likelihood of identifying optimal solutions early in the evolutionary process. Additionally, this approach can reduce computational costs by minimizing unnecessary exploration of less relevant areas. As a result, the algorithm is expected to converge more rapidly toward high-quality solutions while ensuring effective coverage in areas where services are most needed.

4 Experimental Results

The simulation results that were obtained when our optimization model was evaluated using the (1 + 1)-ES algorithm are shown in this section. We report on our experiments and detail the parameters used in our computer simulations.

4.1 Description of Experiments

We evaluated the objective function to compare the impact of the initialization process in the (1 + 1)-ES algorithm on the problem of UAV location for emergency scenarios. The experiment consisted of evaluating 30 different snapshots. In each snapshot, the location of the present mobile users (MU) remained constant, while the number of the requesting mobile users (RMU) varied according to the Erlang traffic. In addition, the RMUs were randomly selected in each snapshot following a uniform distribution. The experiments were made in Python programming language version 3.6 and were run on the CPU of a Jetson TX2, operating under Ubuntu 18.04 LS. The system features 64-bit Denver 2 A57 CPUs, 8 GB of 128-bit LPDDR4 memory with a bandwidth of 58.4 GB/s, and an NVIDIA Pascal GPU equipped with 256 cores.

Table 1 shows the used algorithm parameters in the carried-out experiments. The first column shows the experiment's names; $(1+1)-ES_{\omega_1}$ used the initialization method defined in de Sect. 3.1, $(1+1)-ES_{\omega_2}$ used the initialization method defined in de Sect. 3.2, and $(1+1)-ES_{\omega_3}$ used the initialization method defined in de Sect. 3.3. The second column is the initial value of step size σ used at each experiment. The third column is the adaptation speed of the σ. The step-size σ gives the search scale and generates a perturbation on the multivariate random vector, causing the mutation in the solutions. σ is adapted at each number of iterations z of the algorithm and according to success rate p_s. z defines how many iterations must elapse to adapt σ. We define the maximum number of generations at 500 for the three experiments. These values are set in previous tests of this paper.

Table 2 shows the used instances. The scenario simulation is based in the area corresponding to the town of Tula in Hidalgo, Mexico, where a natural disaster interrupted communication [11]. We consider 8,000 MUs distributed within the

Table 1 Simulation parameters of the (1 + 1)-ES for three experiments

Experiment	Step size σ	Value z
$(1+1)-ES_{\omega_1}$	0.002	20
$(1+1)-ES_{\omega_2}$	0.002	20
$(1+1)-ES_{\omega_3}$	0.002	20

Table 2 Instances that emulate the simulation scenario used in the (1 + 1)-ES evaluation

Description	Value
Affected area	12 × 12 km
Number MUs	8000
Number UAVs	4
Average amount RMUs	2958
Number channels	50

Table 3 Setting parameters for the snapshots that evaluate the (1 + 1)-ES

Description	Value
Carrier frequency	700 MHz
Transmitter power ABS	95 dB
UAVs altitude	122 m
UAV antenna gain	10 dB
RMUs altitude	1.5 m
RMU antenna gain	3 dB

affected area. We consider four UAVs to provide communication services. The average amount RMUs consider the traffic intensity with an outage probability of 1%, and the average time duration of an emergency call at three minutes as reported in [12].

The used network parameters are shown in Table 3. They allow voice transmission in an LTE network operating at the 700 MHz frequency [13]. We assumed the UAVs-ABSs in operation are low-altitude platforms (LAPs) because LAPs are less expensive than high-altitude platforms (HAPs). LAPs are typically used to improve cellular communications [14].

4.2 Results Comparison

Table 4 shows results obtained by evaluating 30 different snapshots for each experiment, i.e., the mean indicates the average PSVS obtained by (1 + 1)-ES. The best solution is the highest PSVS, while the worst solution is the lowest PSVS across

Table 4 Obtained experimental results of 30 different snapshots evaluation in each experiment. *ST D* is the deviation standard

Experiment	Mean	Best solution	Worst solution	STD
$(1+1)-ES_{\omega_1}$	0.6926	**0.7898**	0.5126	0.0950
$(1+1)-ES_{\omega_2}$	0.6898	0.7733	0.5210	0.0749
$(1+1)-ES_{\omega_3}$	0.7403	0.7822	0.6238	0.0419

different snapshots using various initialization methods in (1 + 1)-ES. Finally, the STD indicates the standard deviation provided by each experiment. The $(1+1) - ES_{\omega_3}$ experiment (initialization method of density RMU) produced solutions that were, on average, 4.77% higher than $(1 + 1) - ES_{\omega_1}$ and 5.05% higher than $(1 + 1) - ES_{\omega_2}$. However, all three methods obtained the best solution with a PSVS above 0.77. In other words, the positions of UAVs can offer a probability of establishing a successful network connection for users or emergency operators of higher than 77%. Nevertheless, the $(1 + 1) - ES_{\omega_1}$ experiment obtained the worst solution; 84% lower than $(1 + 1) - ES_{\omega_2}$ and 11.12% lower than $(1 + 1) - ES_{\omega_3}$. This indicates that, in the worst-case scenario, the (1 + 1)-ES algorithm can generate UAV positions that provide a probability of establishing a successful network connection for users or emergency operators of around 51%.

Figure 4 shows the best solution obtained in each evaluated snapshot. Comparing their results, they are similar, which could suggest that the initialization method does not influence the solution search process. As mentioned earlier, the initialization method searches for the diversification of the solutions, even though the (1 + 1)-ES algorithm starts with only a single solution. We examine the evolution of the best solution obtained in each experiment, as shown in Fig. 5.

Figure 5 illustrates that, in some iterations, the algorithm produced a PSVS of zero, indicating that it failed to find a feasible solution during those iterations. On the other hand, the $(1 + 1) - ES_{\omega_1}$ experiment converged prematurely, limiting its exploitation capacity. In contrast, the $(1 + 1) - ES_{\omega_3}$ experiment lost some exploitation capacity but continued to explore the search space. However, the $(1 + 1) - ES_{\omega_2}$ experiment obtains a balance between exploitation and exploration processes. This may be related to the parameter setting of the (1 + 1)-ES algorithm, which increases the diversity in the iteration process of the algorithm.

Finally, Fig. 6 shows the final scenario of each experiment. In this figure, the blue points represent the MUs inside the affected area, the color triangles represent the locations of the UAVs, and the color circles represent the area covered by each UAV.

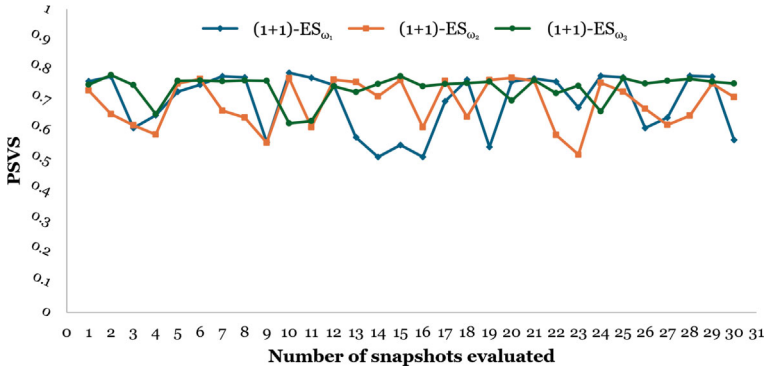


Fig. 4 The best solutions found by (1 + 1)-ES for UAV locations problem of each initialization method

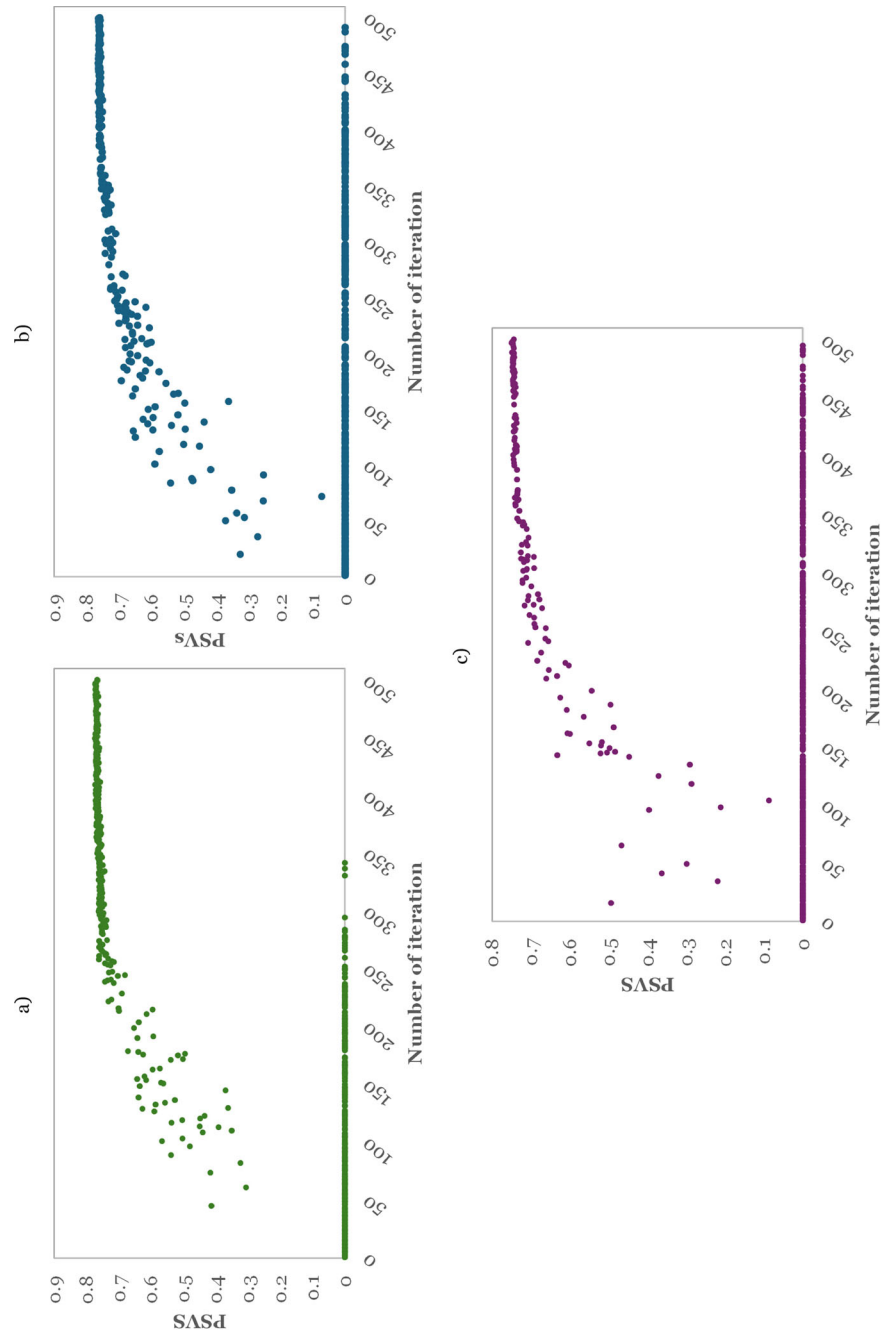


Fig. 5 Evolution process of the best solutions found in each experiment

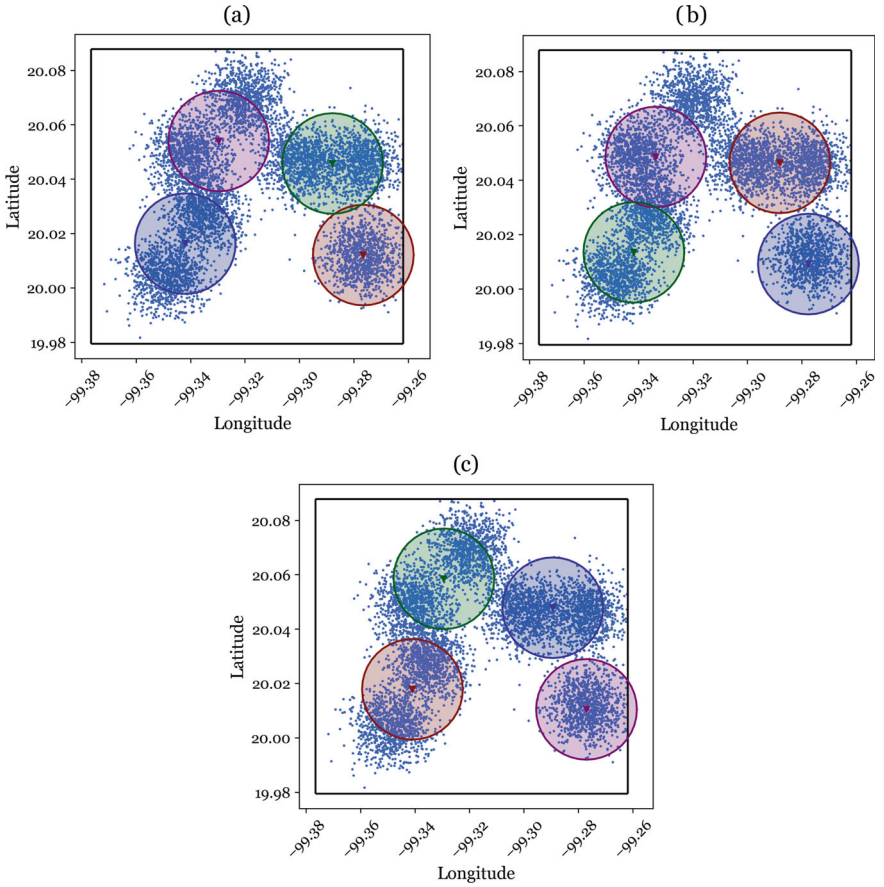


Fig. 6 The best solutions found in each experiment. **a** the best solution obtained for (1 + 1)-ES with the classic random method, **b** the best solution obtained by (1 + 1)-ES with random RMU method, and **a** the best solution obtained by (1 + 1)-ES with density RMU method

Inside the colored circles are the mobile users that were adequately served (i.e., successful services). The location of UAVs is the optimal combination of locations given by (1 + 1)-ES. While there are no noticeable differences in the UAV locations, experiment $(1+1)-ES_{\omega_1}$ (used classical random initialization method) is more dispersed, with a standard deviation of 0.0949 compared to $(1+1)-ES_{\omega_3}$, which has a standard deviation of 0.0419. Therefore, $(1+1)-ES_{\omega_3}$ finds better solutions on average and produces less variable results, even in the worst-case solution.

5 Conclusion and Future Work

In this work, we conducted a comparative analysis of the impact of different initialization methods on the $(1 + 1)$-ES algorithm for the UAV localization problem. While the best results obtained across different snapshots were similar among the methods, the density RMU method ($(1 + 1) - ES_{\omega_3}$) proved to be more robust, producing higher average solution quality with lower variability compared to the classic random method ($(1 + 1) - ES_{\omega_1}$) and the random RMU method ($(1 + 1) - ES_{\omega_2}$). This indicates that the density RMU method provides more reliable outcomes, particularly in scenarios where performance is critical. In addition, in the worst-case scenarios, $(1 + 1) - ES_{\omega_3}$ demonstrated a higher PSVS, highlighting its stability. However, the $(1 + 1) - ES_{\omega_2}$ method maintains a good balance between exploration and exploitation, reducing the risk of premature convergence.

Future work will focus on identifying or developing initialization methods to enhance the algorithm's robustness and ensure it consistently delivers reliable solutions in emergency scenarios, where dependable communication is crucial.

Also, we will explore how quantum computing could enhance the effectiveness of AI-driven optimization strategies, such as those employed in this study. Quantum computing holds the potential to revolutionize the optimization of complex problems, offering advantages in terms of processing speed and the ability to handle vast solution spaces more efficiently. In the context of UAV localization, quantum computing, could provide faster and more accurate solutions by leveraging quantum algorithms that explore multiple possibilities simultaneously, potentially overcoming the limitations of classical methods like premature convergence.

Acknowledgements Gabriela Rodríguez-Cortés gratefully acknowledges the scholarship from CONAHCYT to pursue her graduate studies.

References

1. Sarhani, M., Voß, S., Jovanovic, R.: Initialization of metaheuristics: comprehensive review, critical analysis, and research directions. International Transactions in Operational Research **30**(6), 3361–3397 (2023)
2. Kazimipour, B., Li, X., Qin, A.K.: A review of population initialization techniques for evolutionary algorithms. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 2585–2592 (2014). IEEE
3. Li, Q., Liu, S.-Y., Yang, X.-S.: Influence of initialization on the performance of metaheuristic optimizers. Applied Soft Computing **91**, 106193 (2020)
4. Ghobadi, K., Mahmoudzadeh, H.: Inferring linear feasible regions using inverse optimization. European Journal of Operational Research **290**(3), 829–843 (2021)
5. Kondamadugula, S., Naidu, S.R.: Accelerated evolutionary algorithms with parameterimportance based population initialization for variation-aware analog yield optimization. In: 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1–4 (2016). IEEE

6. Li, Q., Liu, S.-Y., Yang, X.-S.: Neighborhood information-based probabilistic algorithm for network disintegration. Expert Systems with Applications **139**, 112853 (2020)
7. Kacprzyk, J., Pedrycz, W.: Springer Handbook of Computational Intelligence. Springer, Berlin (2015)
8. Bäck, T., Foussette, C., Krause, P.: Contemporary Evolution Strategies vol. 86. Springer, Berlin (2013)
9. Bartz-Beielstein, T., Branke, J., Mehnen, J., Mersmann, O.: Evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **4**, 178–195 (2014)
10. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin (2015)
11. economista, E.: Telmex reporta caídas de Infinitum en 18 poblaciones de Hidalgo por las intensas lluvias. https://www.eleconomista.com.mx/empresas/Telmex-reporta-caidas-de-Infinitum-en-18-poblaciones-de-Hidalgo-por-las-intensas-lluvias-20210908-0135.html (2021)
12. información, C.: Estandarización de los servicios de llamadas de emergencia a través del número único armonizado 9-1-1 (Nueve, Uno, Uno). https://www.gob.mx/911/documentos/norma-tecnica-para-la-estandarizacion-de-los-servicios-de-llamadas-de-emergencia-a-traves-del-numero-unico-armonizado-9-1-1 (2018)
13. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press, New York, USA (2005)
14. Fotouhi, A., Qiang, H., Ding, M., Hassan, M., Giordano, L.G., Garcia-Rodriguez, A., Yuan, J.: Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. IEEE Communications surveys & tutorials **21**(4), 3417–3442 (2019)

SDG 15 Life on Land

An Intelligent System Design for Automated Quality Control of Agricultural Produce Based on Computer Vision

Kenia Picos, Ulises Orozco-Rosas, and Kevin Ruiz

Abstract Currently, the agricultural industry exports a large number of fruits and vegetables, which require a good classification and selection for quality control. This process is commonly carried out manually, where workers visually verify if the product has defects and the appropriate size, shape, and color for its final packaging. Although it is carried out semi-automatically in some industries, this is the most traditional way of exporting in the country. The aforementioned entails a considerable duration in production and the risk of not meeting quality specifications. Therefore, this work proposes the design of an efficient system that is capable of automatically selecting and validating agricultural produce in a more controlled manner. In this research, we focus on the inspection of cucumbers using computer vision techniques for the classification of shape, size, and color in an automated manner. Experimental tests are carried out with artificial intelligence tools. These experiments are carried out on-site using a single high-range industrial camera. The proposed system is evaluated using objective metrics yielding high efficiency for practical applications in agricultural produce inspection.

Keywords Product inspection · Computer vision · Artificial intelligence · Agriculture applications · Deep learning

K. Picos (✉) · U. Orozco-Rosas · K. Ruiz
CETYS Universidad, Tijuana, B.C., Mexico
e-mail: kenia.picos@cetys.mx

U. Orozco-Rosas
e-mail: ulises.orozco@cetys.mx

K. Ruiz
e-mail: kevinr@cetys.edu.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

O. H. Montiel Ross et al. (eds.), *Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1*, Studies in Computational Intelligence 1200,
https://doi.org/10.1007/978-3-031-85614-3_26

1 Introduction

Cucumber production is one of the most important agricultural contributions in Mexico. In 2020, Mexico ranked number 7 in world production and number 1 in Latin America [1]. With a total annual production of 1 Million tons of cucumber per year, of which around 80% is used for export, mainly to the United States. In terms of volume, cucumber is one of the vegetables with the highest consumer demand. The country has implemented conditions to grow it all year round to maintain the demand for cucumber in the market, making Mexico the largest exporter of fresh cucumber [2]. Cucumbers are mainly eaten raw. Nevertheless, they can be processed to produce value-added products, for example in juices, pickled/fermented, skin care products, cosmetics, and scents [3].

Product inspection is a crucial task that has a great impact on agricultural productivity, which can be improved not only by applying modern agricultural processes but also with computerized inspection tools for accurate product classification and defect detection [4].

Monitoring agricultural products is of utmost importance to control quality, especially if they are for export, and increasingly artificial vision technology is immersed in this problem to achieve automatic inspection solutions [5]. Because humans cannot control visual inspection in a wide range of actions, such as detecting defects at the speed of a conveyor belt, nor can they quantitatively measure color, shape, or volume by sight, computer vision plays a crucial role.

Image processing is used to enhance features, and correct, or manipulate information within the input scene. Some image processing techniques can be used to extract relevant information based on features of the input image [6]. Since the input scene captures the cucumber surface, features may not be very distinctive. In some tasks, this is difficult to reach with image processing techniques or barely the human eye. In this case, deep learning architectures produce higher accuracy in image inspection than human visual inspection [7].

Image classification using artificial intelligence has been demonstrated to have several advantages over basic image feature extraction techniques in terms of classification accuracy. Currently, artificial intelligence techniques have been applied to accelerate the defect classification process with considerable precision in various agricultural products such as tomatoes, squash, grapes, and apples, among others [8]. In the particular case of cucumbers, we have challenges in classifying their shape, color, and size to meet certain required quality standards, while they move on a conveyor belt on-site. Omer et al. [4] focused on the early detection of cucumber disease by using deep learning with YOLOv5 architecture [9, 10]. Their study created a cucumber leaf dataset presenting different diseases for analysis. There is still a lack of dataset which includes different problems of cucumber since shape, texture, color, pests, etc.

As we can notice, these techniques depend on image conditions provided in the dataset [11]. Balanced classes in the dataset are crucial for image classification performance. However is very common to deal with imbalanced classes in producing

datasets due to human pre-selection in the field. In [11] demonstrates how imbalanced training data can affect the results of the artificial neural network. In this work, we proposed a method for balancing classes with data augmentation by using image processing techniques. An obvious problem is the type and quantity of captured examples, of false and true classes, because it was carried out through a previous selection at the pickers. Machine learning provides us with an accurate product classification even when applying data augmentation techniques.

In this work, we propose an intelligent system for automatic quality control of agricultural products. The work focuses on the problem of quality inspection of cucumbers, in which the following strategies are addressed:

- An intelligent system based on artificial vision techniques using convolutional neural networks for cucumber classification is designed.
- Data augmentation strategies are proposed for an enriched source of examples for the artificial intelligence network.
- A proposal is made for the measurement of the size and color of the cucumber.
- A volumetric inspection of the product using a multi-camera arrangement is proposed.

The rest of the paper is organized as follows. Section 2 presents the theoretical background of image classification for product quality verification and image techniques for shape inspection. In Sect. 3, we explain the proposed method for automated quality control of agricultural produce based on an artificial intelligence model and computer vision techniques. Section 4 presents the experimental results of performance evaluations for on-site cucumber inspection. Three different experiments were carried out: cucumber shape classification, size and color measurement, and cucumber volumetric inspection. The conclusions and future work are discussed in Sect. 5.

2 Fundamentals

Convolutional neural networks (CNN) are designed to process multi-dimensional array data. The main use of CNNs in image applications is detection, classification, and segmentation [12]. A CNN architecture is mainly composed of several layers: convolutional layers, pooling layers, and densely connected layers [13]. The last layer is known as the output layer, which contains the final nodes of the prediction.

A convolutional layer consists of a convolution between an input feature map from the previous layer $\mathbf{x}^{l-1}$ and a 2D filter containing weights $\mathbf{w}^{l}$ [14]. This operation is performed on a sequence of layers with a non-linear transformer σ, typically denoted by a linear rectification unit (ReLU) [15] for each element, as follows in Eq. (1) [14].

$$\mathbf{x}_j^l = \sigma\left(\sum_{i=1}^{N^{l-1}} \mathbf{x}_i^{l-1} * \mathbf{w}_{i,j}^l + b_j^l\right) \tag{1}$$

where a convolution between the N^{l-1} input feature maps $\mathbf{x}_i^{l-1}$ and the corresponding filter $\mathbf{w}_{i,j}^{l}$, plus a bias term b_j^l is represented. ReLU operation is depicted as Eq. (2) [16].

$$\sigma(x) = \max\{x, 0\} \tag{2}$$

After each convolutional layer is applied, an oversampling/undersampling process, known as pooling, is performed [17]. A series of densely connected layers are used at the end of a set of convolutional layers, each convolutional layer is followed by a pooling layer.

The squared difference between the true value and the prediction is used to measure the model's accuracy to be close to the expected values [18].

$$L_n = (t_n - f(x_n; w_0, w_1))^2 \tag{3}$$

The above expression indicates how much accuracy is being lost by using f to model t. In Eq. (3), L_n indicates the loss, x is an input, and t is an output defined as a function that depends on the input as $t = f(x)$. The values w_0 and w_1 produce a straight line resulting from a regression, which ideally passes close to all the points in the data set. If we want to know the value of the loss across the entire data set, let's consider the average loss shown in In Eq. (4) [18].

$$L = \frac{1}{N}\sum_{n=1}^{N} L_n(t_n - f(x_n; w_0, w_1)) \tag{4}$$

To find the best values for w_0 and w_1 that allow the lowest loss (the lower the loss, the better the result) we can apply the *argmin* operation to find the argument that minimizes the function, as follows in Eq. (5).

$$\underset{w_0, w_1}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N} L_n(t_n - f(x_n; w_0, w_1)) \tag{5}$$

Different CNN architectures allow object or pattern classification in images. A fundamental framework established by LeCun et al. in 1998 developed a multilayer neural network known as LeNet-5 [19], which was initially applied to classify handwritten digits. Today, this same network architecture can be taken for general object classification applications [20]. LeNet-5 is a convolutional neural network architecture that contains convolutional layers, a pooling process, and fully connected layers in the final stage [21], as shown in the diagram in Fig. 1.

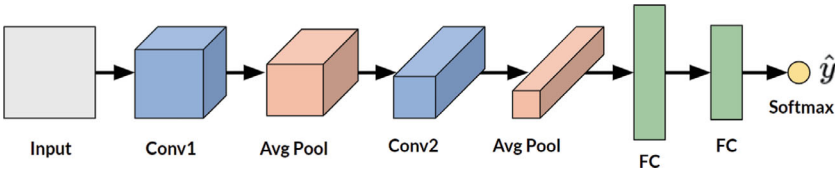


Fig. 1 Convolutional neural network, known as Lenet-5 architecture, that is applied to image classification

3 Proposed Methodology

A practical methodology for automated on-site quality control of cucumbers is addressed. The proposed method is composed of three steps, which are in the following subsections. First, a shape classification is implemented to decide which product may be eligible for export. In the case of cucumber, we build our database from the location of multiple cameras on a conveyor belt at the site of an agricultural produce facility. This decision-making task uses machine learning techniques by proposing a custom convolutional neural network for the cucumber quality control approach. Second, we proposed a system capable of measuring the visual attributes of the moving cucumber, which are size and color. For this, we place an industrial camera in the top view to inspect several product samples. This implementation provides a wide information for decision-making by using image processing techniques. Finally, the third step consists of a new design for volumetric inspection of cucumbers. This implementation includes a multi-camera approach for multi-view reconstruction. Color and texture information contributes to the generation of a 3D model, under controlled conditions. Volumetric inspection enriches the information for defect detection, improving the visual inspection that has traditionally been performed.

3.1 Cucumber Shape Classification

The proposed methodology for cucumber classification is described in this section. Here, we replace the visual inspection made on-site with an automatic shape classification system. The proposed method is composed of three main steps: input normalization, data augmentation, and the convolutional neural network model. Figure 2 illustrates these steps, which are explained below.

Step 1: Input Images

In this step, the images are captured in real time by an industrial camera. This camera is located at the top of a conveyor belt, where the agricultural product is transported

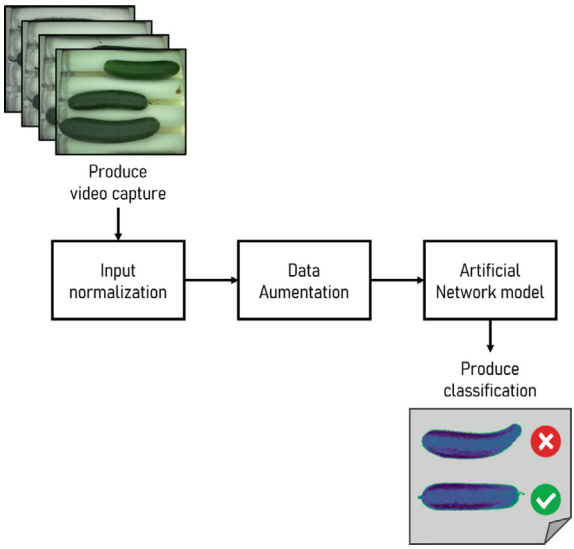


Fig. 2 Block diagram of the proposed methodology for cucumber classification

for final packaging. The captured scene is the upper view of the produce over the conveyor belt. In the case of cucumber, the product remains held between the belt rollers, making it possible to capture the separated product. In our input images, more than one cucumber can appear in the scene, as shown in Fig. 3.

The captured images are stored in the cloud. First, the images are loaded from the cloud to be processed. Then, the input images are normalized in size (500×140 pixels). Also, the images are normalized in intensity values with a float type from 0 to 1. Some basic threshold is applied to extract the background. Last, the images are labeled within two classes: curved (Class 1) and non-curved (Class 0).

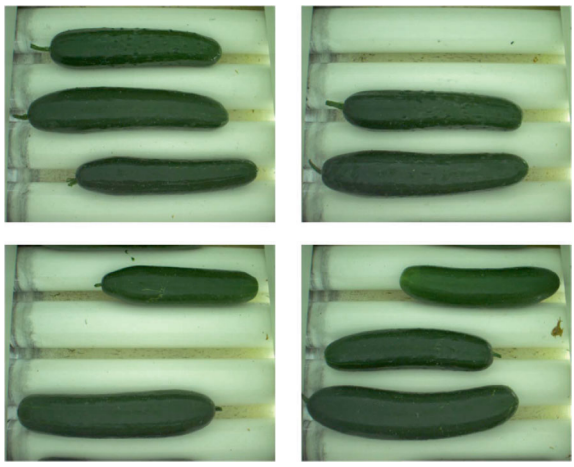


Fig. 3 Sample of input images captured on-site by an industrial camera for cucumber inspection

Step 2: Data Augmentation

In this step, an algorithm for creating different versions of the original video set is defined. Data augmentation is employed to obtain a wide amount of images and balance each class [22]. Generally, in agriculture applications such as produce inspection, it is very common to have a data set with true class examples due to a prior selection by farmers. However, it is difficult to obtain false class produce images, due to the great diversity of product failures, some due to non-visible shapes, color, and texture that are not possible to capture with the industrial camera at the speed of the conveyor belt. In this step, we use data augmentation to achieve two thousand images for each class. Data augmentation is applied for both classes. From an original image input we randomly implemented different image operations [23], such as horizontal reflection, vertical reflection, and scaling, as shown in Fig. 4 for the curved class and Fig. 5 for the non-curved class. The new synthetic images are concatenated in the data set as an input.

Step 3: Convolutional Neural Network Model

A convolutional neural network model is proposed to perform the classification task. In our approach, we distinguish two classes of cucumber by its shape, in which Class 1 and 0 indicate a curved and non-curved form, respectively.

In the architecture design, we use LeNet's model as a basis, as described in Sect. 2. From this, we adjust and enlarge in deep the model configuration to adapt it to the cucumber inspection problem by using a different architecture between convolutional

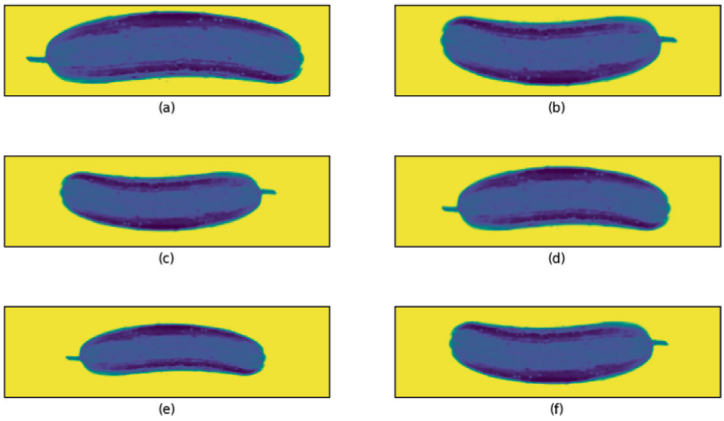


Fig. 4 Data augmentation of the curved class from an original image input with random functions to apply horizontal flip, vertical flip, and scaling. **a** Original image. **b–f** Augmented images using scaling, rotation, and reflection from the original image

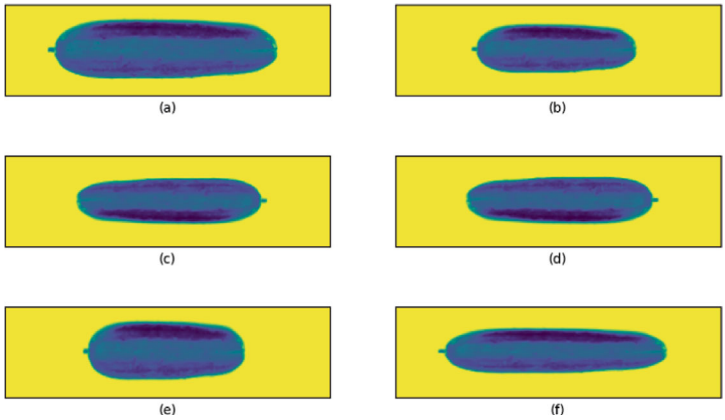


Fig. 5 Data augmentation of the non-curved class from an original image input with random functions to apply horizontal flip, vertical flip, and scaling. **a** Original image. **b–f** Augmented images using scaling, rotation, and reflection from the original image

layers. We introduce the data set distributed with 64% of data for training, 16% of data for validation, and 20% of data for testing.

The proposed convolutional neural network model stacks some convolutional layers, employing a rectified linear unit (ReLU) as an activation function in each one. Each convolutional layer or pair of convolutional layers is followed by a max pooling layer. At the end of the model, a fully connected neural network with two hidden layers is employed for the classification task. Finally, the last layer contains the output, 2 classes: curved and non-curved. In Fig. 6 the proposed model for cucumber classification is illustrated.

At the first convolutional layer, we define 64 large filters (7×7), we employ a stride of one due to the input images that have been cropped and resized to 250×70 pixels from the data augmentation process. The input images contain a single-color channel, i.e., grayscale.

Then, we add a max-pooling layer that employs a pool size of two, to divide each spatial dimension by a factor of two. After that, we repeat the same configuration twice: two convolutional layers with small filters (3×3) followed by a max-pooling layer. The number of filters is incremented by a factor of two as we raise the con-

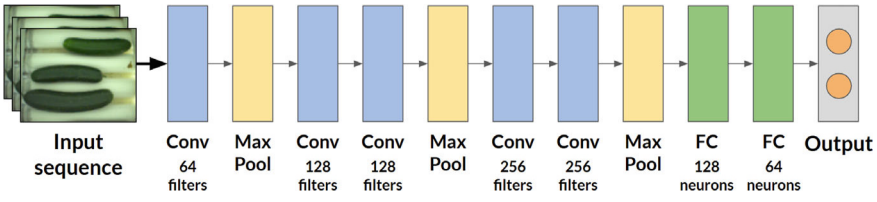


Fig. 6 Proposed model architecture for cucumber classification

volutional neural network to the output layer, we start with 64 filters, then 128 in the second stage, and finally with 256 filters in the third stage. The main reason to increase the number of filters is due to it is a common practice after each pooling layer since the number of low-level features is often fairly low, but it can be combined with higher-level features [15].

At the final stage, we employ a fully connected network that is composed of two hidden dense layers with 128 and 64 neurons each. Due to it being a classification task with two classes, the output layer has two units, and it uses the Softmax [24] activation function.

3.2 Cucumber Size and Color Measurement

Size and color attributes are of utmost importance for the inspection of agricultural products [25]. Image processing techniques are extensively used for automatic quality control of produce in the external size and color. This yields an external quality inspection applying conversion between pixels and physical dimensions [26]. Some examples are shown in Fig. 3 in which you can see images of the cucumber passing through a conveyor belt. Some of them are affected by lighting which causes the color measurement to be inaccurate.

A methodology is designed with artificial vision techniques to automate the measurement of the cucumber inside the conveyor belt to increase the quality of inspection [27]. These steps are designed for measuring the size and color of the cucumber, where we describe the methodological proposal, the proposed block diagram for size and color measurement is illustrated in Fig. 7, which consists of the following steps.

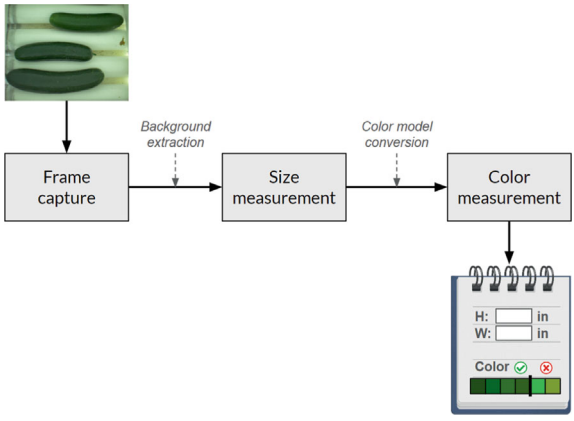


Fig. 7 Block diagram of the proposed methodology for cucumber size and color measurement

Step 1: Input Frame

Input frames are captured from a camera placed upper the conveyor belt. The frame contain more than one cucumber over the conveyor rollers, as shown in Fig. 3. These images are normalized and the background is extracted to segment the region of interest of the object. Some image filtering is required when motion blur is presented due the displacement of the conveyor.

Step 2: Size Measurement

For size estimation of the cucumber, we first need to extract the background. As seen in Fig. 3 the conveyor rollers are yet visible. A binary threshold is applied to the input image to distinguish the cucumber area. In Algorithm 1 we find the left upper and the right lower corners to establish the delimitation of the area of the cucumber from the binary image I_b. Length (L) and width (W) measurements are computed as the following rule:

$$L, W = (left\ upper\ corner - right\ lower\ corner)\ band\ dim/image\ shape$$

It is taken as a reference that the conveyor belt measures 18 in. wide, and the image width is 640 pixels. Each pixel equals $18/640 = 0.028125$ inches with this rule.

Algorithm 1 Automatic procedure for size measurement of the cucumber.

Require: Input image I, image size $imshape$, and conveyor belt size b
Ensure: Cucumber size in inches: length (L) and width (W)

```
function SIZE MEASUREMENT(I)
    N, M ← imshape
    I_b ← Binary Threshold(I,threshold)
    I_b ← 255 − I_b
    for i ← 1 to N do
        for j ← 1 to M do
            if I_b(i, j) > 0 then
                X, Y ← X.append(i), Y.append(j)
            end if
        end for
    end for
    X_0, Y_0 ← min(X),min(Y) Gives the left upper corner
    X_1, Y_1 ← max(X),max(Y) Gives the right lower corner
    L ← (X_1 − X_0)b/M
    W ← (Y_1 − Y_0)b/M
    return L, W
end function
```

Table 1 Tolerance for color measurement in the HSV color model

	Range	Tolerance
H	0.25–0.35	0.3 ± 0.05
S	0.45–0.75	0.6 ± 0.15
V	50–120	85 ± 35

Step 3: Color Measurement

The color measurement process is straightforward. Due to the input image being affected by various lighting conditions and motion blur, the color measurement is for reference only. From the binary image I_b, the object can be extracted by multiplying it by the input image. Remember that image I_b contains a value of 1 within the object area and 0 otherwise. This way we will only process the color values of the cucumber.

$$I_h = (I)(I_b) \tag{6}$$

We compute the histogram from RGB values of the resulting image I_h, as shown in Eq. (6). The range of color for acceptance is very subjective and depends on the requirements of the consumer. However, we proposed a range of values in the HSV color model as guidance, as follows in Table 1.

3.3 Cucumber Volumetric Inspection

A methodology for volumetric inspection based on multi-view image reconstruction is proposed in this section. The volumetric inspection system is designed from multiple image captures, image processing, and 3D reconstruction, as described below.

Step 1: Multi-camera Capture

In this step, the observed 3D scenario is captured by a camera array, in which the input images contain a target or multiple targets represented by multiple views. In this work, we explain an example with stereo images and our proposed system with multiple camera arrays, which process four simultaneous cameras for image capture.

Step 2: Image Processing

Due to image capture being performed using a set of cameras, an optoelectronic device is needed to digitize and store multiple images simultaneously. For these, an

electronic board is used that supports up to four cameras to multiplex the capture and processing of images with an electronic device or mini PC. In this step, image filtering and alignment are used.

Step 3: 3D Reconstruction

Finally, the processed images from Step 2 are used to calculate the disparity map and generate a resulting point cloud for the 3D reconstruction algorithm. This step produces the visualization of the reconstructed scene given by multiple views of the target using camera arrays.

As an example, when the input images consist of a stereo pair, the reconstruction can be performed by calculating the stereo disparity. First, it is necessary to calculate a matching algorithm, such as the semi-global matching (SGBM) algorithm. Then, the depth map is obtained by adding absolute differences (SAD) between both images [28]. A projection image is converted to 3D information that produces the resulting points with a three-dimensional coordinate space and a Red-Blue-Green (RGB) three-color channel. A summary algorithm of reconstructing a 3D scene from stereo images is represented by Algorithm 2. This algorithm produces a PLY output file by converting the final data, which contains a set of points, each with 6 parameters: 3 columns for coordinates in 3D space and 3 columns for the RGB information of each color channel.

Algorithm 2 Cucumber reconstruction from pair of images.

Require: Input images I_1, I_2, image size $imshape$, and focal distance f
Ensure: Reconstructed scene point cloud `outputfile.py`
$window \leftarrow$ Define window size
$mindisp \leftarrow$ Disparity min value
$stereo \leftarrow$ Stereo matching($window$, $mindisp$)
$disp \leftarrow$ Stereo disparity(I_1, I_2)
$\mathbf{Q} \leftarrow \mathbf{Q}$ matrix
$points \leftarrow$ 3D image reprojection($disp$, $\mathbf{Q}$)
$mask \leftarrow disp > min(disp)$
$points \leftarrow$ Calculate color points($mask$)
$data \leftarrow$ Generate output 3D model in PLY file

The current method introduces a simultaneous image capture of four images with different views of the scene. A set of key points between images is computed using a feature-matching algorithm that produces the fundamental matrix estimate. The camera pose can be estimated by calculating the fundamental matrix and key points [29]. Here, the estimation can be optimized by minimizing the residual of the reprojection vector. The graphs generated from matching points can then be triangulated and adjusted to merge all graphs. The summarized pseudocode of a 3D scene reconstruction from multi-camera arrays is described in Algorithm 3. As we can see, the presented algorithm produces dense graph reconstruction processing to

generate output point cloud data with 3D coordinates and RGB color channels in PLY format.

Algorithm 3 Cucumber reconstruction from multicamera array.

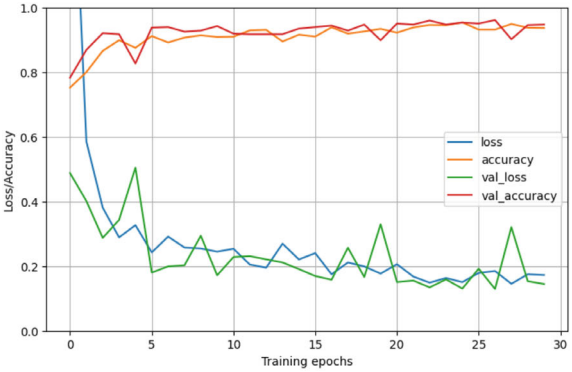
Require: Input images I_1, I_2, I_3, I_4, image size $maxsize$, and focal distance f
Ensure: Reconstructed scene point cloud `outputfile.py`
  $graphlist \leftarrow$ List of input images
  for image list length **do**
    $keypoints \leftarrow$ Compute SIFT algorithm
    $\mathbf{F} \leftarrow$ Compute fundamental matrix($keypoints$)
    $\mathbf{K} \leftarrow$ Compute matrix K
    $\mathbf{E} \leftarrow \mathbf{K}^T \mathbf{F}_0 \mathbf{K}$
    $campose \leftarrow$ Compute camera pose($\mathbf{E}$, $\mathbf{K}$, $keypoints$)
    $graphlist \leftarrow$ Generate graphics information($campose$, $\mathbf{K}$, $keypoints$, f)
    $graphtriang \leftarrow$ Compute Mesh triangulation($graphlist$)
    $graphadjust \leftarrow$ Compute graphic adjust($graphtriang$)
  end for
  $graphmerge \leftarrow$ Compute graphics merge
  for image list length **do**
    $graphlist \leftarrow$ Dense graphics processing(image list, $maxsize$)
  end for
  for image list length **do**
    $graphlist \leftarrow$ Dense reconstruction processing($graphmerge$, $\mathbf{K}$, $maxsize$)
  end for
  $data \leftarrow$ Generate output 3D model in PLY file

4 Experimental Results

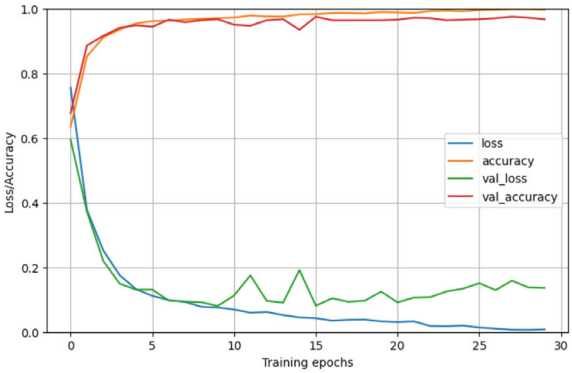
In this section, the experimental results obtained in the design of the proposed system for automatic quality control of agricultural products are presented. First, we will present the results of the cucumber classification system using an artificial intelligence system, and we will evaluate the results, as well as the comparisons of different architectures to test the system. Following this development, we will present the results of the measurement of the color and size of the cucumber using an embedded multi-camera vision system. Finally, we will carry out computational experiments for volumetric inspection with three-dimensional models obtained from object reconstruction techniques.

4.1 Shape Classification

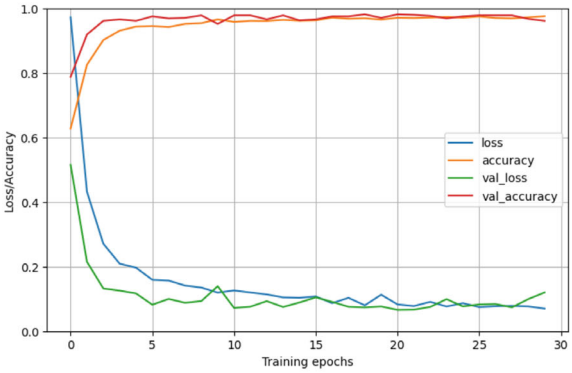
For shape classification three different neural networks were employed, a deep neural network (DNN), a LeNet-based convolutional neural network (LeNet CNN), and a proposed convolutional neural network (CNN). Figure 8 shows the learning curves in



(a)



(b)



(c)

Fig. 8 Learning results in terms of loss and accuracy of **a** DNN architecture, **b** LeNet CNN architecture, and **c** the proposed CNN architecture

Table 2 Loss and accuracy results with the test set

Metric	DNN	LeNet CNN	Proposed CNN
Loss	0.1604	0.1941	**0.1207**
Accuracy	0.9449	0.9649	**0.9662**

terms of loss and accuracy for the training process using the training and validation set. The learning curves display the accuracy and loss values respectively versus the epoch that are the number of times the fitting algorithm passes through the training set.

In the results shown in Fig. 8 for the three different architectures, we can observe that the training accuracy and the validation accuracy steadily increase during training, while the training loss and the validation loss decrease. Therefore, we can observe good results for the three variants, but we also have to consider some particular observations. In Fig. 8a we can observe that the learning curves for the DNN are too unstable but relatively close to each other from the first epoch to the last. Table 2 shows that the DNN obtained a final loss of 0.1604 and an accuracy of 0.9449 with the test set representing a quite good result. The test set is composed of 20% of the full data set, i.e., 800 test images.

Figure 8b presents the results for the LeNet CNN architecture, we can observe that the validation curves are close to the training curves at first, but they get further apart over time mainly in the loss curves, which shows that there is a little bit of overfitting. The training set performance ends up beating the validation performance, as is generally the case when we train for long enough [15]. Table 2 shows that the LeNet CNN obtained a loss of 0.1941 and an accuracy of 0.9649 representing a good result.

The results for the proposed CNN are shown in Fig. 8c, the validation curves are relatively far apart from each other at first, but they get further closer over time, which shows a stable training process. For accuracy, we can observe that the training and validation curves overlap almost perfectly at the end of training. Table 2 shows that the proposed CNN obtained a loss of 0.1207 and an accuracy of 0.9662 representing the best results obtained over the other two architectures presented.

A binary classification task is assigned in the three architectures implemented, the DNN, the LeNet CNN, and the proposed CNN. Therefore, the classifier can make two errors: incorrectly assigning a curved class to a non-curved cucumber or incorrectly assigning a non-curved class to a curved cucumber. The confusion matrix shown in Fig. 9 displays the overall results for each architecture.

The classification results for the DNN architecture are shown in Fig. 9a. From the results, we can determine that this architecture presents a 1.10% misclassification rate, i.e., $(42 + 2)/4000 = 0.0110$. A 10.37% false positive rate, i.e., the fraction of negative examples that are classified as positive $(42/405 = 0.1037)$. And a 0.50% false negative rate, i.e., the fraction of positive examples that are classified as negative $(2/395 = 0.0050)$. From Fig. 9b we can observe that the LeNet CNN obtains a 0.70%

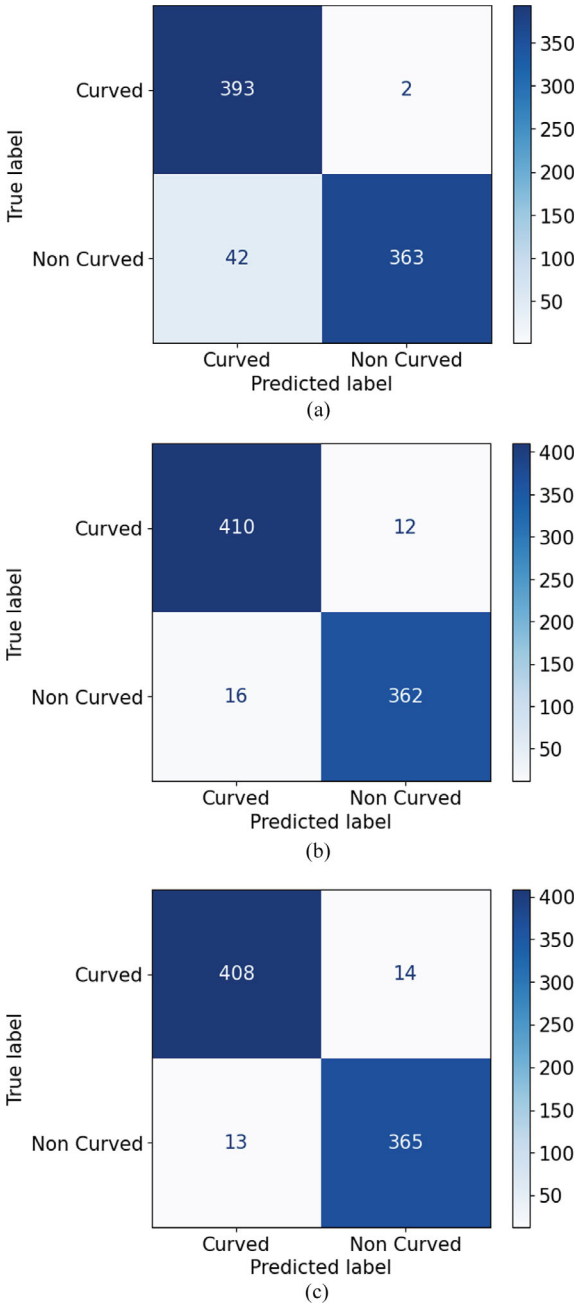


Fig. 9 Comparison between predicted and true values for both classes of **a** DNN architecture, **b** LeNet CNN architecture, and **c** the proposed CNN architecture

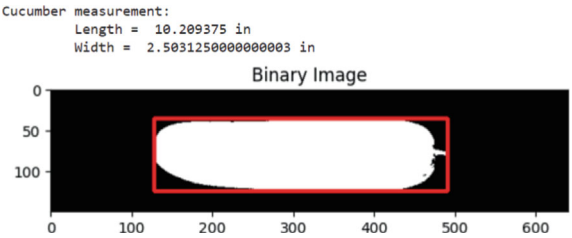


Fig. 10 Result of the automatic measurement of the cucumber using pixels to inches conversion

misclassification rate, i.e., $(12 + 16)/4000 = 0.0070$. A 3.20% false positive rate, i.e., $12/374 = 0.0320$. And a 3.75% false negative rate, i.e., $16/426 = 0.3750$. The best classification results were obtained from the proposed CNN architecture in terms of misclassification. In Fig. 9c we can observe a 0.67% misclassification rate, i.e., $(14 + 13)/4000 = 0.0067$. A 3.69% false positive rate, i.e., $14/374 = 0.0369$. And a 3.18% false negative rate, i.e., $13/408 = 0.0318$.

4.2 Size and Color Measurements

The proposed system performs a frame capture from an industrial camera, as shown in Fig. 7, in which the cucumbers are in motion over a conveyor band. Adequate lighting is very important for this measurement since the condition of shadows or textures can be present in the image causing false positives. After a background extraction, the program binarizes the image, as shown in Fig. 10. The binary image is obtained by thresholding to make it easier to find the region of interest (ROI) of the object. This searches through the pixels until it finds a change to register at the start and end corners of the object. This difference is calculated to find the distance as follows in Eq. (7).

$$l = \Big[p_1(x, y) - p_0(x, y)\Big]\frac{L_{in}}{L_{pix}} \tag{7}$$

where l is the distance in inches between starting $p_0(x, y)$ and finishing $p_1(x, y)$ points of the ROI. Here, the size of the image in pixels and the dimensions of the captured scene in inches are $L_{pix} = 640$ pixels and $L_{in} = 18$ inches, respectively. The ratio L_{in}/L_{pix} is the size in inches for each pixel.

Color measurement was performed from the region of interest (ROI) of the input image, where 10 samples were taken from each cucumber to help subjective decision-making on the quality of the product's appearance. Table 3 shows the results of an example of color measurement for a single cucumber. The results are measured as being within or outside the specifications proposed in Table 1. A general color inspection is performed by computing the histogram of the input image in three color channels, as shown in Fig. 11.

Table 3 Example of color measurement for each cucumber in ten samples

Sample	R	G	B	H	S	V	Color	Spec
1	75	124	34	0.2574	0.7258	124		Out
2	58	94	38	0.2738	0.5957	94		In
3	60	93	46	0.2836	0.5053	93		In
4	56	91	49	0.3055	0.4615	91		In
5	50	75	36	0.2735	0.5200	75		In
6	28	59	23	0.3101	0.6101	59		In
7	59	85	42	0.2674	0.5058	85		In
8	18	53	20	0.3428	0.6603	53		In
9	72	93	34	0.2259	0.6344	93		Out
10	96	130	46	0.2352	0.6488	130		Out

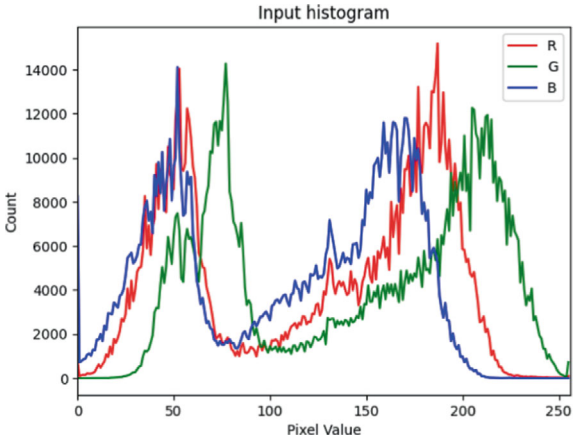


Fig. 11 Result of the automatic measurement of the cucumber using pixels to inches conversion

4.3 Volumetric Reconstruction

The developed system performs a reconstruction of the given scene from the proposed multi-camera array system, as shown in Fig. 12. First, the system simultaneously captures four different views of the scene provided by the camera array. As illustrated in the system in Fig. 12, we use four input images, each representing a different view of the scene [30]. The images shall contain a scene of the conveyor belt with a cucumber being inspected.

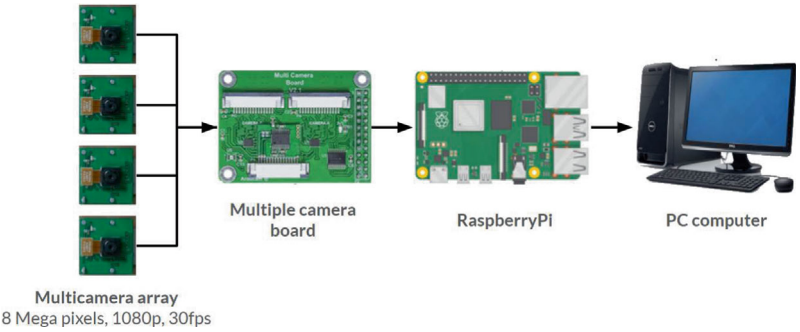


Fig. 12 Proposed diagram for implementing 3D digital reconstruction of produce using multi-camera arrays

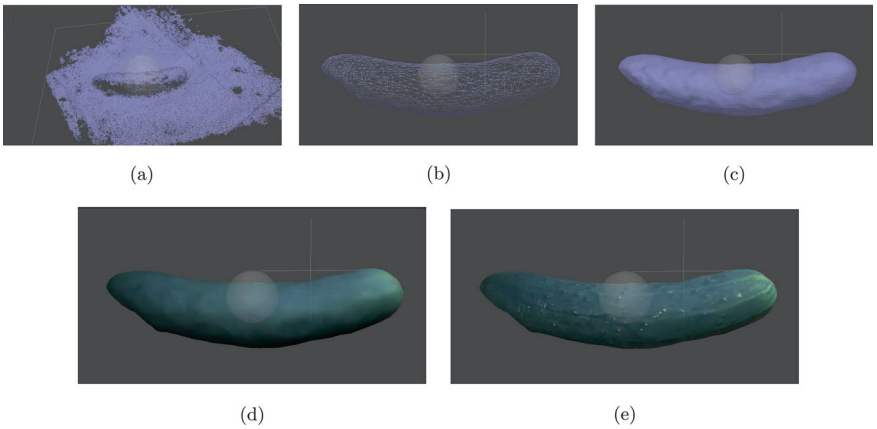


Fig. 13 Result of volumetric inspection in five stages: **a** Point cloud, **b** Wireframe, **c** Solid surface, **d** Low frequency, and **e** High frequency colored texture

We implement the Algorithm 2 in a Python programming code. The program produces an accurate reconstruction from the given input images into a point cloud as an output file in PLY format. The different stages of digital reconstruction are illustrated in Fig. 13. The resulting point cloud of the inspected surface is shown in Fig. 13a. In Fig. 13b, triangulation is applied to build the wireframe mesh. The solid outer surface of the mesh is calculated in Fig. 13c, and finally each face is associated with a color and texture value, as shown in Fig. 13d still at low frequency and in Fig. 13e with high-frequency texture as the final result. The 3D model obtained from the reconstruction is stored as a PLY file providing the 3D coordinate information of each point, in which, each vertex is associated with color information in three RGB channels. The experiments were carried out with the following computational resources. Platform: Windows, CPU: 12th Gen Intel(R) Core(TM) i7-12700H, CPU family: 6 model: 154 signature: 906A3h, RAM: 15.7 GB, OpenGL Vendor: NVIDIA

Corporation, OpenGL Renderer: NVIDIA T1200 Laptop GPU/PCIe/SSE2, OpenGL Version: 4.6.0 NVIDIA 538.59. The final cucumber Mesh resulted in the following characteristics. Point cloud: 133,899 points, Faces: 1928, Vertices: 1023, Maximum Model Size: 131 KB.

5 Conclusions

An efficient algorithm is proposed for automated quality inspection of agricultural products using artificial vision. In this work, we focused on cucumber inspection in terms of shape, size, and color to detect defects. The proposed algorithm was tested in real scenarios within several imaging conditions such as background clutter, light, and multiple views of the tested product. The current methodology is reliable to implement in real applications for the inspection of different produce. The proposed system is performed by using artificial intelligence with computer vision techniques by using 2D and 3D images. This system automatically classifies cucumber characteristics such as area, length, shape, and defects for quality inspection, helping us detect them correctly for decision-making.

In the experimental results, we show that the quality of the produce can be classified into two classes: curved and not curved. The cucumber shape inspection was performed by using artificial intelligence networks yielding a high performance in classification results. Also, the length and width indicate that the cucumber has reached a certain growth, consequently, a product that is tender and smaller in size will generally have a higher acceptance. Likewise, color measurements have been performed to achieve satisfaction levels. These measurements are carried out so that the final product achieves a good perception for export.

In this work, we also employed 3D reconstruction for volumetric inspection by using the multi-view approach, which yields good performance and high reliability in decision-making in classification. However, there is room for improvement in the volumetric inspection system, starting from the speed of image capture from an industrial conveyor belt. In future work, different camera locations will be tested to improve the reconstruction. Likewise, the acquisition of a greater number of images that show varieties of defects, and also including them in a more robust artificial intelligence network. In general terms, this system presents robustness in agricultural applications. The proposed algorithm demonstrates an efficient performance for automatic quality inspection in cucumber classification which can be implemented to inspect a wide variety of produce.

References

1. Resultados definitivos del Censo Agropecuario 2022–2023. INEGI. Comunicado de prensa número 667/23 (2023). https://www.inegi.org.mx/app/saladeprensa/noticia.html?id=8571
2. A short guide to Mexican cucumber production and export. ProducePay (2022). https://producepay.com/blog/a-short-guide-to-mexican-cucumber-production-and-export/
3. Paramithiotis, S. (ed.): Lactic Acid Fermentation of Fruits and Vegetables. CRC Press, London, England (2021)
4. Omer, S.M., Ghafoor, K.Z., Askar, S.K.: Lightweight improved yolov5 model for cucumber leaf disease and pest detection based on deep learning. Signal Image Video Process. **18**(2), 1329–1342 (2024)
5. Jana, S., Parekh, R., Sarkar, B.: Automatic classification of fruits and vegetables: A texture-based approach. In: Algorithms in Machine Learning Paradigms. Studies in computational intelligence, pp. 71–89. Springer, Singapore (2020)
6. Hou Ming Chong, Yap, X.Y., Chia, K.S.: Effects of different pretrained deep learning algorithms as feature extractor in tomato plant health classification. Pattern Recognition and Image Analysis **33**(1), 39–46 (2023)
7. Throop, J.A., Aneshansley, D.J., Anger, W.C., Peterson, D.L.: Quality evaluation of apples based on surface defects: development of an automated inspection system. Postharvest Biology and Technology **36**(3), 281–290 (2005). https://doi.org/10.1016/j.postharvbio.2005.01.004
8. Bhargava, A., Bansal, A., Goyal, V.: Machine learning–based detection and sorting of multiple vegetables and fruits. Food Anal. Methods **15**(1), 228–242 (2022)
9. Hussain, M.: YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision (2024). https://arxiv.org/abs/2407.02988
10. Liu, H., Sun, F., Gu, J., Deng, L.: Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode. Sensors **22**(15) (2022). https://doi.org/10.3390/s22155817
11. Das, S.: A new technique for classification method with imbalanced training data. Int. J. Inf. Technol. **16**(4), 2177–2185 (2024)
12. Lopez-Montiel, M., Orozco-Rosas, U., Sánchez-Adame, M., Picos, K., Montiel, O.: Evaluation of deep learning algorithms for traffic sign detection to implement on embedded systems. In: Recent Advances of Hybrid Intelligent Systems Based on Soft Computing. Studies in computational intelligence, pp. 95–115. Springer, Cham (2021)
13. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition **77**, 354–377 (2018). https://doi.org/10.1016/j.patcog.2017.10.013
14. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. International Journal of Computer Vision **128**(2), 261–318 (2020)
15. Geron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd edn. O'Reilly Media, Inc., Sebastopol, California, United States (2019)
16. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, Massachusetts, United States (2016). http://www.deeplearningbook.org
17. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013). https://doi.org/10.1109/TPAMI.2013.50
18. Rogers, S., Girolami, M.: A First Course in Machine Learning, 2nd edn. CRC Press, London, England (2020)
19. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791

20. Lopez-Montiel, M., Orozco-Rosas, U., Sanchez-Adame, M., Picos, K., Ross, O.H.M.: Evaluation method of deep learning-based embedded systems for traffic sign detection. IEEE Access **9**, 101217–101238 (2021)
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
22. Bhuse, P., Singh, B., Raut, P.: Effect of data augmentation on the accuracy of convolutional neural networks. In: Joshi, A., Mahmud, M., Ragel, R.G., Thakur, N.V. (eds.) Information and Communication Technology for Competitive Strategies (ICTCS 2020), pp. 337–348. Springer, Singapore (2022)
23. Hong, Y., Chen, Y.: PatchMix: patch-level mixup for data augmentation in convolutional neural networks. Knowledge and Information Systems **66**(7), 3855–3881 (2024)
24. M., C., Hugh: Deep Learning, 1st edn. Springer, Cham, Switzerland (2023)
25. Zhang, K., Chen, X., Wang, H.: Research on external quality inspection technology of tropical fruits based on computer vision. In: Tavana, M., Patnaik, S. (eds.) Recent Developments in Data Science and Business Analytics, pp. 165–174. Springer, Cham (2018)
26. Unay, D., Gosselin, B.: Stem and calyx recognition on 'jonagold' apples by pattern recognition. Journal of Food Engineering **78**(2), 597–605 (2007). https://doi.org/10.1016/j.jfoodeng.2005.10.038
27. Ramirez-Paredes, J.-P., Hernandez-Belmonte, U.-H.: Visual quality assessment of malting barley using color, shape and texture descriptors. Computers and Electronics in Agriculture **168**, 105110 (2020). https://doi.org/10.1016/j.compag.2019.105110
28. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
29. Hu, Q., Luo, J., Hu, G., Duan, W., Zhou, H.: 3d point cloud generation using incremental structure-from-motion. Journal of Physics: Conference Series **1087**(6), 062031 (2018). https://doi.org/10.1088/1742-6596/1087/6/062031
30. Ruiz, K., Picos, K., Orozco-Rosas, U.: Analysis of three-dimensional object reconstruction algorithms based on multi-camera arrays. In: Awwal, A.A.S., Iftekharuddin, K.M., Diaz-Ramirez, V.H., Márquez, A. (eds.) Optics and Photonics for Information Processing XIV, vol. 11509, p. 115090. SPIE, San Diego, California (2020). https://doi.org/10.1117/12.2568788. International Society for Optics and Photonics